

NUMÉRO 555 FÉVRIER 1994

DOMESTICUS : BORNIER DE SORTIES TOR

LA RADIO NUMÉRIQUE ET LE HSP 50016

LA TRANSMISSION VIDÉO SUR PAIRE TORSADÉE

INTERFACE DE PUISSANCE POUR MOTEURS C.C.

CINQ OSCILLATEURS DE WIEN POUR LE TEST

T 2438 - 555 - 24,00 F

Sommaire

ELECTRONIQUE APPLICATIONS

MENSUEL édité par
PUBLICATIONS GEORGES VENTILLARD
S.A. au Capital de 5 160 000 F
2 à 12, rue de Bellevue, 75019 PARIS
Tél. : 42.00.33.05 - Fax : 42.41.89.40
Télex : 220409 F
Principaux Actionnaires :
- M. Jean-Pierre Ventillard
- Mme Paule Ventillard

Président-Directeur-Général,
Directeur de la Publication :
Jean-Pierre VENTILLARD
Directeur de la Rédaction :
Bernard FIGHIERA
Rédacteur en Chef :
Claude DUCROS
Marketing/Ventes :
Jean-Louis PARBOT
Tél. : 42.00.33.05.
Création maquette :
Rachid MARAI
Inspection des Ventes :
Société PROMEVENTE - M. Michel IATCA
11, rue de Wattignies - 75012 PARIS.
Tél. : 43.44.77.77 - Fax : 43.44.82.14.
Publicité :
Société Auxiliaire de Publicité
70, rue Compans, 75019 PARIS
Tél. : 42.00.33.05
C.C.P. PARIS 37 93 60
Directeur commercial : Jean-Pierre REITER
Chef de publicité : Francine FIGHIERA
Assistée de : Laurence BRESNU
Abonnement : Marie-Christine TOUSSAINT
Voir notre tarif «spécial abonnement».
Pour tout changement d'adresse, envoyer la dernière bande accompagnée de 2,80 F en timbres.
IMPORTANT : ne pas mentionner notre numéro de compte pour les paiements par chèque postal.

Ce numéro a été tiré à 35 600 exemplaires
Dépôt légal février 94 - éditeur 1739
Mensuel paraissant en fin de mois.
Distribué par S.A.E.M. Transports-Presse
Photocomposition COMPOGRAPHIA - 75011 PARIS -
Photo de couverture : E. Malemanche.









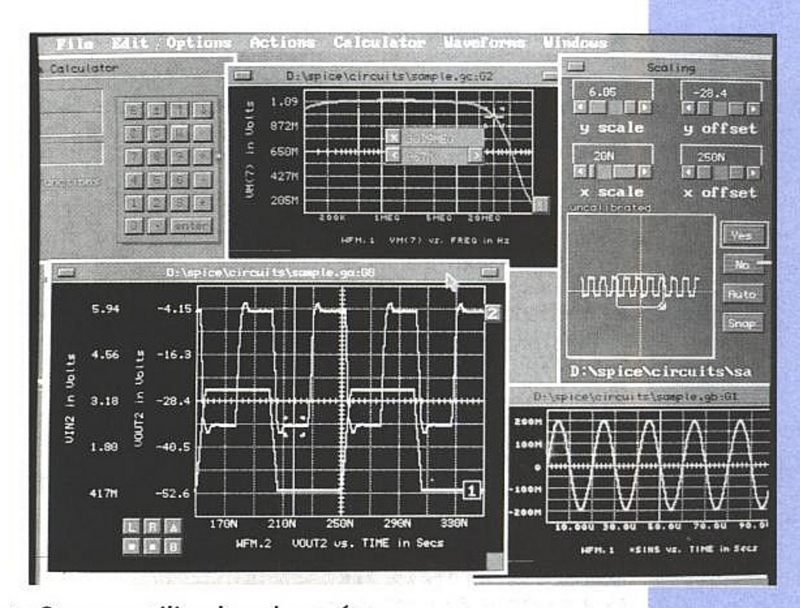
Numéro 555 - Février 1994

ISSN 1144-5742

Ont participé à ce numéro : J. Alary, P. de Carvalho, F. de Dieuleveult, C. Djabian, J. Garbay, A. Garrigou, P. Gueulle, S. Landerretche, P. Morin, P. Oguic, D. Paret, T. Riflart, J.-L. Vern.



Is SPICE EN ACQUISITION DE DONNÉES

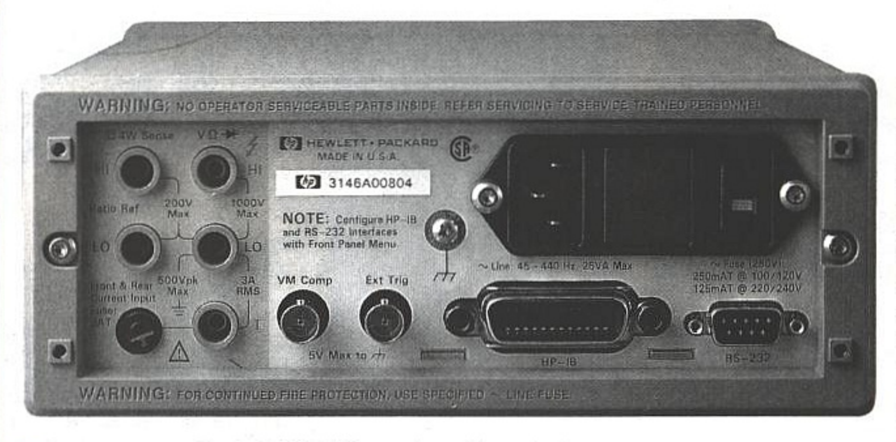


Le logiciel de simulation IsSpice de la société américaine Intusoft exploite, pour visualiser les résultats de ses calculs, un fichier sauvegardé en mode ASCII. Ce fichier comporte tous les éléments assurant la compatibilité avec le format imposé à l'origine par l'université de Berkeley. Le module de visualisation graphique IntuScope utilise les données stockées dans ce fichier texte afin de les afficher à l'écran sous forme de courbes. Cette méthode offre ainsi à l'utilisateur la possibilité de créer un fichier de données décrivant un système réel, et ainsi de profiter de tous les outils mathématiques mis à sa disposition dans IntuScope pour traiter ses résultats (FFT, diagrammes de Bode, mesures RMS, formes polynomiales ...). Enfin, ce logiciel permet de numériser les courbes présentes à l'écran et de les transformer en sources indépendantes que l'on pourra ensuite introduire dans une simulation SPICE.

L'acquisition de données

Lorsque l'on développe un dispositif électronique, le cahier des charges impose certaines caractéristiques auxquelles le prototype doit se plier : bruit généré, précision de la grandeur délivrée, dérive en continu ... Si un simple relevé d'oscilloscope permet d'apprécier rapidement la forme d'un signal de sortie, il n'autorise pas, en revanche, des mesures précises telles la linéarité d'une pente, ou encore le contenu fréquentiel d'un bruit. Dans la plupart des cas, le concepteur devra se rabattre sur un système d'acquisition couteux afin de traiter ses résultats à l'aide de l'outil informatique. De nombreux programmes existent dans ce domaine et conduisent à un traitement efficace des résultats de mesure. Citons par exemple LabWindows de National-Instruments, VEE de Hewlett-Packard ...
La société Intusoft commercialise depuis plusieurs années le logiciel de simulation IsSpice qui, combiné au programme de visualisation graphique IntuScope et la saisie de schéma SpiceNet, offre un atelier de simulation SPICE très complet. Les lecteurs désireux de mieux connaître ce logiciel pourront consulter l'article paru à son sujet dans Electronique Radio-plans (voir bibliographie) ou encore, demander une disquette de démonstration à la société Excem (78 MAULE) qui assure la distribution du programme en France. IntuScope met en oeuvre un véritable oscilloscope logiciel incorporant de nombreuses fonctionnalités que l'on ne trouve que sur certains équipements haut de gamme : calcul de FFT's, fenêtres d'apodisation de type Hanning, Cosinus, diagrammes de Bode, de Nyquist, bases de temps multiples ... Il serait donc particulièrement intéressant de bénéficier de toutes ces caractéristiques afin d'effectuer du traitement d'informations sur le système réel que son concepteur souhaite tester. Lors du déroulement de la simulation, IsSpice produit un fichier de type ASCII, dans lequel il range une multitude de données. Nous verrons plus bas comment celles-ci sont agencées. La méthode retenue consiste à acquérir des variables (tension,





■ *Le panneau arrière du HP 54401 ayant servi à nos tests.*

courant, fréquence ...) par l'intermédiaire d'un bus IEEE-488 ou par le biais d'une carte d'acquisition. Le programme les stocke ensuite sur disque selon le format réclamé par IntuScope. Il suffit de lancer ce dernier et de lui faire ingérer les données enregistrées. Toute l'architecture d'IntuScope s'ouvre alors au concepteur afin d'analyser en détail les performances de son dispositif. De plus, IntuScope pouvant sauvegarder des images aux formats variés (TIF, EPS ...), la présentation des rapports de mesure en sera grandement améliorée. Le paragraphe "Utiliser IntuScope" décrit brièvement comment lancer et travailler avec IntuScope sur votre fichier utilisateur.

SIMULATION ET RANGEMENT DES CALCULS

Le fichier de sauvegarde porte obligatoirement l'extension .OUT et rassemble les résultats des calculs décrivant, pas à pas, les potentiels de chacun des noeuds choisis par l'utilisateur. En comparaison des logiciels concurrents, IsSpice offre une flexibilité supplémentaire quant à la sauvegarde des résultats. Par exemple, il est possible d'affecter un nombre de points précis aux courbes qui seront affichées et ainsi accélérer leur traitement sous IntuScope. En pratique, 1000 points permettent une étude confortable sans pour autant mobiliser à outrance la mémoire du PC. Des valeurs plus faibles autorisent le chargement d'un nombre de courbes supérieur. Le calcul des FFT's s'en trouve amélioré puisque l'échantillonnage s'effectue sous fréquence fixe et constante. On veillera, dans ce dernier cas, à retenir un nombre d'échantillons correspondant à une puissance binaire (128, 1024 ...). D'autre part, il est possible d'intervenir sur le pas de simulation et forcer IsSpice à travailler avec une incrémentation temporelle adaptée à l'étude. Par défaut, si l'on ne spécifie aucune valeur pour la variable TMAX, IsSpice choisit le pas maximum égal à (TSTOP - TSTART) divisé 50. Le fait de stipuler un intervalle entre échantillons permet d'optimiser le temps de calcul en fonction de la précision réclamée par le dispositif simulé : inutile de choisir un pas de 1ns si l'on souhaite tester un comportement transitoire d'une dizaine de millisecondes.

L'étude transitoire s'effectue par le biais du mot clé .TRAN suivi de quatre paramètres dont deux sont optionnels (TSTART et TMAX). Voici quelques exemples illustrant ce que nous venons d'énoncer précédemment :

.TRAN TSTEP TSTOP TSTART TMAX

.TRAN 1NS 1US ; effectue une simulation durant 1µs et stocke les données toutes les 1ns. On obtient 1µs/1ns soit 1000 points par courbe.

.TRAN 1NS 1US 10NS 10NS ; effectue une simulation de 1000 points en stockant une valeur chaque nanoseconde mais ne sauvegarde les résultats qu'après 10ns de calcul. Empêche IsSpice de prendre un pas supérieur à 10ns.

Description du fichier de données

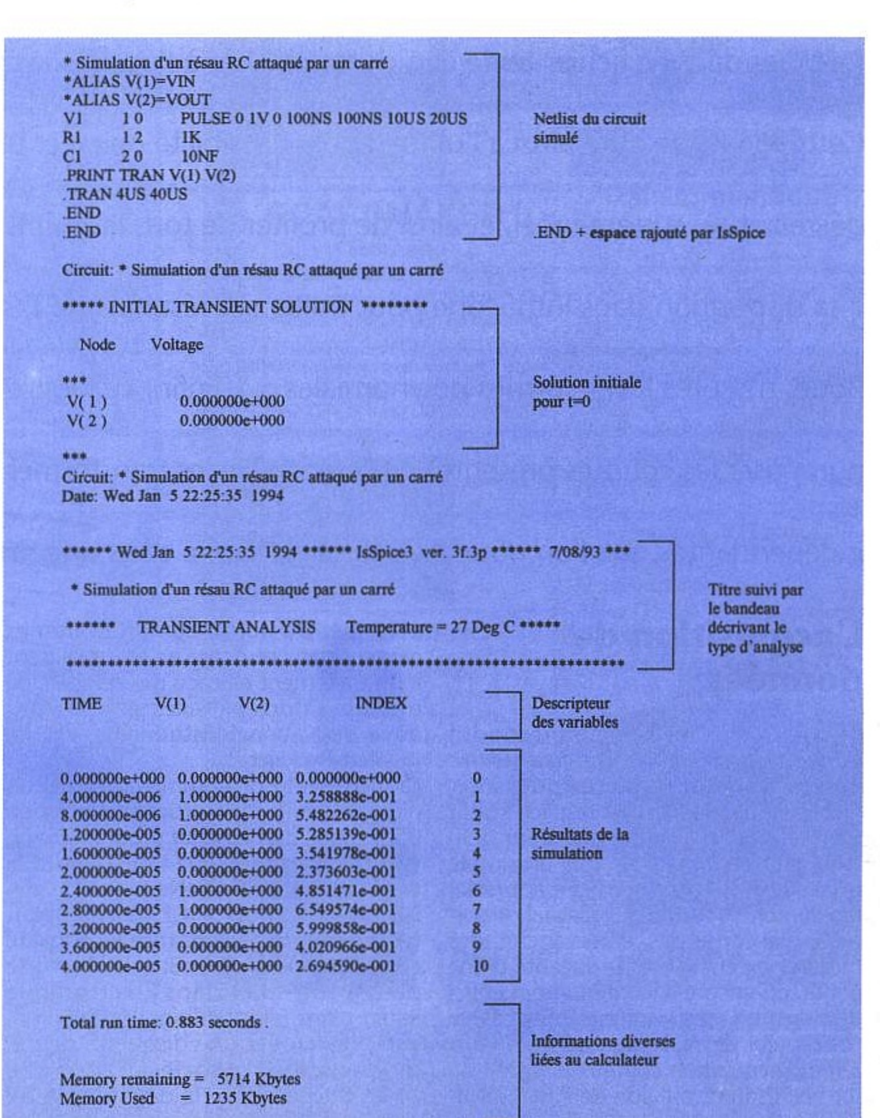
Après ces quelques remarques, il est temps de passer à la constitution d'un fichier écrit par IsSpice. Fort heureusement, les lignes clé nécessaires à la création d'un fichier utilisateur sont nettement moins nombreuses que celles produites par IsSpice qui elles, collent parfaitement au standard de Berkeley. Le fichier EXEMPLE.CIR que nous avons simulé représente un réseau RC attaqué par une source de signaux carrés :

```
* Simulation d'un réseau RC attaqué par un carré
V1   1 0   pulse 0 1V 0 100nS 100nS 10uS 20uS
R1   1 2   1K
C1   2 0   10nF
.PRINT TRAN V(1) V(2); sauvegarde les noeuds 1 et 2
.TRAN 4µs 40µs ; stocke 1 point toutes les 4µs ; durant 40µs soit 10 points au total
. END
```

Le résultat de simulation se trouve stocké dans le fichier EXEMPLE.OUT et son contenu se trouve en **figure 1**, associé aux commentaires descriptifs que nous avons rajoutés.

Créer son fichier utilisateur

La documentation jointe au logiciel IsSpice décrit avec précision la syntaxe et sa position à respecter par l'utilisateur lorsqu'il créera son propre fichier. L'exemple donné en **figure 2** inclut tous les éléments indispensables à la compatibilité avec le standard SPICE de Berkeley. En fait, IntuScope fait preuve d'une grande souplesse lorsqu'il charge un fi-



```
* Simulation d'un résau RC attaqué par un carré
*ALIAS V(1)=VIN
*ALIAS V(2)=VOUT
V1      1 0     PULSE 0 1V 0 100NS 100NS 10US 20US
R1      1 2     1K
C1      2 0     10NF
.PRINT TRAN V(1) V(2)
.TRAN 4US 40US
.END
.END
                                   Netlist du circuit simulé
                                   .END + espace rajouté par IsSpice

Circuit: * Simulation d'un résau RC attaqué par un carré

***** INITIAL TRANSIENT SOLUTION ********

  Node       Voltage

***
 V( 1 )          0.000000e+000
 V( 2 )          0.000000e+000
                                   Solution initiale pour t=0
***
Circuit: * Simulation d'un résau RC attaqué par un carré
Date: Wed Jan  5 22:25:35  1994

****** Wed Jan  5 22:25:35  1994 ****** IsSpice3  ver. 3f.3p ****** 7/08/93 ***

 * Simulation d'un résau RC attaqué par un carré

******    TRANSIENT ANALYSIS      Temperature = 27 Deg C *****

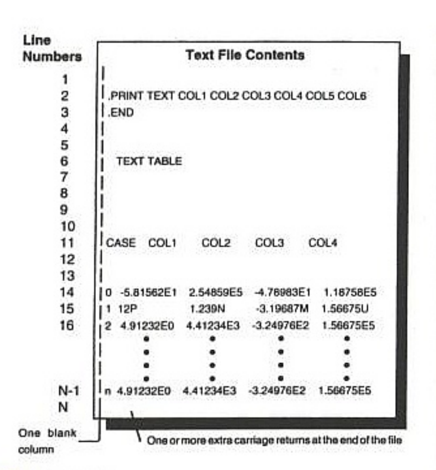
*******************************************************************
                                   Titre suivi par le bandeau décrivant le type d'analyse

TIME          V(1)          V(2)               INDEX
                                   Descripteur des variables

0.000000e+000  0.000000e+000  0.000000e+000    0
4.000000e-006  1.000000e+000  3.258888e-001    1
8.000000e-006  1.000000e+000  5.482262e-001    2
1.200000e-005  0.000000e+000  5.285139e-001    3
1.600000e-005  0.000000e+000  3.541978e-001    4
2.000000e-005  0.000000e+000  2.373603e-001    5
2.400000e-005  1.000000e+000  4.851471e-001    6
2.800000e-005  1.000000e+000  6.549574e-001    7
3.200000e-005  0.000000e+000  5.999858e-001    8
3.600000e-005  0.000000e+000  4.020966e-001    9
4.000000e-005  0.000000e+000  2.694590e-001    10
                                   Résultats de la simulation

Total run time: 0.883 seconds .

Memory remaining =   5714 Kbytes
Memory Used      =   1235 Kbytes
                                   Informations diverses liées au calculateur
```



■ *Figure 2*

```
.PRINT TRAN V(1)
.END

***** TRANSIENT ANALYSIS *****

TIME            V(1)            INDEX

0.000000        0.56290         0
0.000001        0.84030         1
0.000002        0.53490         2
0.000003        0.16130         3
0.000004        0.01460         4
0.000005        0.41190         5
0.000006        0.75730         6
0.000007        0.61880         7
0.000008        0.45570         8
0.000009        0.09280         9
```

■ *Figure 3*

chier de données. Après divers essais, dont les conclusions ont été confirmées par Intusoft, nul n'est besoin d'insérer autant de lignes et blancs comme l'indique la **figure 2**. En fait, il suffit simplement d'indiquer en majuscules, avec un .PRINT placé en début de fichier, le nom des variables disponibles ainsi que le type d'étude auxquelles elles se rapportent : TRAN ou AC par exemple. Juste après, IntuScope s'attend à détecter le mot clé .END. Attention, il faut **impérativement** placer un blanc (0x20 en hexadécimal) après le D et ensuite insérer un retour chariot. En l'absence de ce caractère, IntuScope affiche le message "There is no data for the selected analysis". Bien que le logiciel puisse accéder aux données, il déroule le fichier en entier afin d'essayer, en vain, de détecter ce .END associé à son espace. Si votre fichier contient plusieurs milliers de points, cette opération peut mobiliser le PC quelques minutes. Après consultation d'Intusoft, il ne s'agit nullement d'une bogue dans le logiciel, mais simplement d'une syntaxe imposée à l'origine par Berkeley et à laquelle IntuScope se conforme. Vous rencontrerez ce même type de désagrément si vous tentez de lire un fichier .OUT généré par PSpice dans le lequel Microsim n'a pas rajouté de blanc après le .END. Comme le dit la documentation d'IntuScope, à la lecture de ce fichier .OUT, vous obtiendrez un message d'erreur similaire à celui énoncé quelques lignes auparavant. Le fait de rajouter avec un éditeur de texte un blanc après le D du .END, jugule cette anomalie. EDIT de DOS (4.0 et au dessus) convient pour cette opération (la taille du fichier est cependant limitée), par contre, malgré son énorme capacité à gérer des lignes, l'EDI de Borland est incapable d'insérer un blanc avant un retour chariot ! Intusoft pense

```
/*
      Programmation du HP34401 pour creer un fichier IntuScope
      IsSpice.c, Christophe BASSO Janvier 1994.
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>

#include "hp34401a.h"              /* Declaration des fonctions du HP34401A */
#include "ibicerr.h"        /* Prototypes des fonctions de gestion d'erreur */

#define path_file ".\\isspice.out"

void main(void)
{
   int      sample=512,                          /* Nombre d'echantillons */
            retard=0,                         /* Periode d'echantillonage */
            loop;
   FILE     *dat_file;                   /* Pointeur du fichier de donnees */
   char     tampon[8800];                /* Stockage des lectures du 34401 */
   float    valeur,
            time,
            step=0.0007;

   clrscr();

/* ---------- Ouverture du fichier de donnees --------------------------- */

   if ((dat_file = fopen(path_file,"wt")) == NULL )
   fail ("Cannot open destination file");

/* ---------- Ecriture de l'entete IntuScope ---------------------------- */

   fprintf (dat_file,"%s\n",".PRINT TRAN V(1)");
   fprintf (dat_file,"%s\n\n",".END ");
   fprintf (dat_file,"%s\n\n","***** TRANSIENT ANALYSIS *****");
   fprintf (dat_file,"%s\t\t%s\t\t\t%s\n\n","TIME","V(1)","INDEX");

/* -------------------- Boucle de lecture/stockage de donnees ------------ */

   init_34401(sample,retard);           /* Initialise et declenche le 34401 */
   read_34401(tampon,sample*17);           /* Recupere les donnees stockees */
   for (loop=0; loop < sample; loop++)
   {
      time = step*loop;
      sscanf (&tampon[(loop)*16],"%f",&valeur);
      gotoxy(2,2);
      printf ("HP34401 = % f \n",valeur);
      gotoxy(2,3);
      printf ("Sample n° %d",loop);
      fprintf(dat_file,"%6.4f\t\t % 6.4f\t\t\t%d\n",time,valeur,loop);
   }
   end_34401();
   fclose (dat_file);
   printf ("\nProgram terminated\n");
}
```

```
/*
     Routines de gestion du multimètre HP34401A
     HP34401A.C, Christophe BASSO 1994
*/

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>

#include "hp34401a.h"              /* Déclaration des fonctions du HP34401A */
#include "ibicerr.h"
#include "d:\language\gpib-pc\decl.h"

int Ud;

void init_34401 (int sample,int retard)
{
   char *buf1;

   if ((Ud = ibfind("HP34401")) & ERR) show_err("Ibfind 34401A");

/* ---------- Passe le time-out a 10 secondes --------------------------- */

   ibtmo (Ud,10000);
   show_err ("ibtmo");

/* ---------- Envoie la séquence d'initialisation du HP34401A------------ */

   buf1 = "*RST";
   ibwrt (Ud,buf1,strlen(buf1));
   show_err ("ibwrt");


/* ---------- Remet à zéro tous les mots d'état ------------------------ */

   buf1 = "*CLS";
   ibwrt (Ud,buf1,strlen(buf1));
   show_err("ibwrt");

/* ---------- Configure le HP34401A en volt-mètre DC -------------------- */

   buf1 = "CONF:VOLT:DC 10V,MAX";
   ibwrt (Ud,buf1,strlen(buf1));
   show_err("ibwrt");

/* ---------- Passe en integration 0.02 plc ----------------------------- */

   buf1 = "VOLT:DC:NPLC 0.02";
   ibwrt (Ud,buf1,strlen(buf1));
   show_err("ibwrt");

/* ----------- Programme un filtre rapide ------------------------------- */

   buf1 = "DET:BAND 200";
   ibwrt (Ud,buf1,strlen(buf1));
   show_err("ibwrt");

/* ---------- Programme le déclenchement externe------------------------- */
```

■ *Figure 4 : source complet du programme d'acquisition en turbo C*



supprimer cette particularité, non décrite dans le manuel, lors de prochaines versions d'IntuScope.
Une fois les deux lignes décrites insérées l'une à la suite de l'autre, il convient d'indiquer, toujours en lettres capitales, le type d'échelle à mettre en place selon la simulation adoptée : transient ou AC. Voici un exemple de bandeau descriptif : ***** TRANSIENT ANALYSIS ***** ou ***** AC ANALYSIS *****. La simulation AC convient particulièrement aux relevés de réponses en fréquence d'un système puisqu'elle incorpore la notion de phase (Vp). Dans le cas général, on choisira le bandeau TRANSIENT. Attention, comme le manuel ne le stipule pas, le bandeau TEXTE ne fonctionne pas sous IntuScope. D'ailleurs, cette option du menu ne peut-être sélectionnée. Elle disparaîtra dans les futures versions.
Juste derrière, une ligne décrit le nom des variables tels qu'IntuScope les affichera. Elle débute par la principale, la variable de temps TIME. Ensuite, espacées par des blancs ou des tabulations, les noms des grandeurs mesurées : V(1), V(2), I(V2) ... On peut également ajouter des *ALIAS afin qu'IntuScope utilise un nom plutôt qu'une dénomination purement électrique : VOUT, VCC ... Finalement, il ne manque plus que le pas d'incrémentation temporel ainsi que les résultats de mesure placés en colonnes en dessous du nom qui leur correspond ou simplement espacés par des blancs ou une tabulation. La fin du fichier se termine classiquement par un EOF (^Z). IntuScope tolère des syntaxes variées sur la sauvegarde des chiffres : notation scientifique 1.235E-3 ou classique, 0.001235. La **figure 3** fournit un court exemple fonctionnel de fichier utilisateur qui reprend les recommandations précédentes. Le lecteur pourra le modifier à loisir afin de tester le comportement d'IntuScope et ainsi, adapter son logiciel d'acquisition en conséquence. Une autre solution peut également consister à modifier un fichier contenant des résultats de mesure stockés en ASCII. Par exemple, une recopie d'écran d'oscilloscope ou d'analyseur de spectre. Le programme à écrire en C consiste alors à insérer l'entête IntuScope décrit ci-dessus complété par le nombre de voies disponibles. On rajoute une variable compteur TIME, incrémentée d'un pas d'une mesure à l'autre afin de se rapprocher du modèle donnée en **figure 3**. Le chargement dans IntuScope ne pose alors aucun problème.

```
    bufl = "TRIG:SOUR BUS";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Passe en retard de déclenchement manuel ------------------- */

    sprintf(bufl,"TRIG:DEL %d",retard);
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Programme le nombre d'échantillons à SAMPLE --------------- */

    sprintf(bufl,"SAMP:COUN %d",sample);
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Supprimme l'affichage du multimètre ----------------------- */

    bufl = "DISP OFF";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Passe en mode "Wait for Trigger" -------------------------- */

    bufl = "INIT";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

    delay (500);                                /* Initialisation complete */

/* ---------- Déclenche la conversion du multimètre --------------------- */

    ibtrg(Ud);
    show_err("ibtrg");
}

void read_34401(char * tampon,long int nombre)
{
    char  *bufl;

/* ---------- Demande la sortie de la valeur convertie ------------------- */

    bufl = "FETC?";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Vide le tampon de sortie du HP34401A ---------------------- */

    ibrd (Ud,tampon,nombre);
    show_err("Ibrd");
}

void end_34401(void)
{
    char *bufl;

/* ---------- Emet un beep pour signaler la fin d'acquisition ----------- */

    bufl = "SYST:BEEP";
    ibwrt (Ud,bufl,strlen(bufl));



    show_err("ibwrt");

/* ---------- Rétabli l'affichage du multimètre ------------------------- */

    bufl = "DISP ON";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

/* ---------- Affiche le mot "Termine" ----------------------------------- */

    bufl = "DISP:TEXT \"Termine\"";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

    delay (2000);

/* ---------- Repasse en mode normal ------------------------------------- */

    bufl = "*RST";
    ibwrt (Ud,bufl,strlen(bufl));
    show_err("ibwrt");

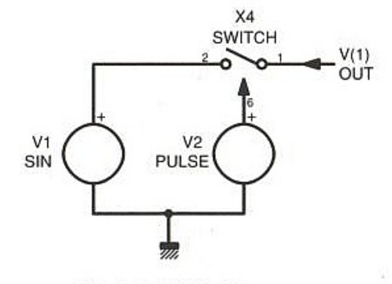
/* ---------- Repasse en mode local -------------------------------------- */

    ibloc (Ud);
    show_err("ibloc");
}
```

■ ***Figure 4 suite***

Développer son programme d'acquisition

Comme énoncé plus haut, de nombreux procédés d'acquisition s'offrent à l'utilisateur pour récupérer des variables dans le but de les traiter. Nous avons, pour notre part, utilisé un voltmètre IEEE de Hewlett-Packard, le HP34401A dont la description détaillée a fait l'objet d'un article dans votre revue (voir bibliographie). Ce dernier supporte le langage SCPI et autorise une fréquence d'échantillonnage maximale proche de la milliseconde (en 4 digits 1/2). Nous avons repris le module développé par l'auteur et publié dans l'article : "La programmation du HP34401A". Il consiste en un ensemble de routines simplifiant l'écriture d'un logiciel d'acquisition de données. Le programme principal effectue alors des appels à ces routines afin de récupérer les valeurs de tension converties par le voltmètre, et les enregistrer au format IntuScope. Le source complet écrit en Turbo C apparaît en **figure 4**. Le fichier d'erreur IBERROR.C correspond à l'ancien source publié dans les articles donnés en bibliographie ; il ne sera pas reproduit ici. Les



V2 : sin 0 1 100Hz 0 0
V1 : pulse 0 1V 0 10us 10us 200us 2ms
X4 : SWITCH SW Ron=0.1 Vt=.5V Vh=.5V
.TRAN 2ms 1023ms 100us

■ ***Figure 5a : modèle d'échantillonneur.***

FFT d'un sinus 100Hz echantillonne a 500Hz

■ *Figure 5b*

FFT d'un sinus 400Hz echantillonne a 500Hz

■ *Figure 5c*

lignes qui intéressent plus particulièrement le lecteur se trouvent commentées. Elles concernent essentiellement l'ouverture du fichier et l'insertion de l'entête IntuScope. On remarquera le caractère blanc rajouté lors de l'écriture du .END dans le fichier. Peu de remarques ensuite, exceptée la sauvegarde du fichier avec une extension .OUT sans laquelle IntuScope ne pourrait sélectionner son nom à l'ouverture.

Attention à la variable TIME qui sera enregistrée. En effet, selon les mesures que vous souhaiter réaliser, elle devra non seulement décrire avec précision le pas séparant deux prises d'échantillon, mais également être constante sur toute la durée de la mesure (temps réel). Pour cela, trois méthodes existent : la première consiste à récupérer l'horloge du PC à chaque échantillon par la fonction Turbo C **gettime**. Il s'agit ensuite de soustraire le temps de l'échantillon n+1 de celui arrivé précédemment (échantillon n). Cette opération renouvelée à chaque valeur conduira au pas d'incrémentation temporel recherché. Malheureusement, la tolérance et surtout le manque de résolution de l'horloge du PC conduirait à des acquisitions peu précises dans le temps. En fait, certaines mesures, comme des dérives continues étalées sur plusieurs heures, se soucient peu d'une faible résolution. En conséquence, il suffira de rajouter un délai de 1s (**delay** (1000) ou **sleep** (1)) ou plus avant chaque échantillon et d'augmenter d'autant la variable TIME.

L'autre méthode se rapproche nettement plus du vrai temps réel. Elle utilise la fonction TRIGGER EXTERNE du voltmètre HP34401A. Ainsi, il suffira d'injecter un signal TTL de fréquence connue et fixe afin de déclencher périodiquement l'instrument de mesure. Si la fréquence d'échantillonnage pilotant le HP34401A vaut précisément 50Hz, alors le pas séparant chaque échantillon sera de 20ms, valeur dont on incrémentera la variable TIME. Le travail en SRQ devient alors obligatoire. Malheureusement, le transfert de données via l'IEEE prend un temps non négligeable (attente du SRQ, envoi du serial polling ...) et réduit considérablement la vitesse d'exécution. De plus, il est relativement difficile d'évaluer avec précision le temps qui s'écoule entre l'arrivée du signal de déclenchement et son acquisition réelle par le PC via le contrôleur IEEE. Cependant, pour des fréquences de répétition lente, on pourra heureusement négliger cette inconnue et la considérer comme un jitter intervenant lors de la prise d'échantillon.

L'exemple de la **figure 4** utilise plutôt la mémoire interne du HP34401A. Dans l'exemple proposé, le voltmètre se charge tout seul de la prise des données et le PC récupère ensuite le contenu de sa mémoire par le biais de l'IEEE. Plus de difficultés liées au temps d'échantillonnage incertain, puisque celui du 34401 peut-être ajusté. Le fichier de configuration 34401.C décrit l'ensemble des instructions à envoyer au multimètre pour obtenir une fréquence de répétition la plus élevée possible. Nous sommes arrivés à environ 700µs, valeur dont TIME se trouve incrémenté. Pour forcer le multimètre à travailler aussi rapidement, il faut supprimer le retard après l'arrivée du top de synchro, baisser la résolution à 4 digits 1/2, utiliser le filtre rapide et enfin, éteindre l'afficheur. Le rafraîchissement de ce dernier perturbe considérablement la prise d'échantillon en ajoutant un important jitter.

Le source IsSpice.C se limite au stockage maximal prévu par le HP, soient 512 échantillons. Cette valeur convient parfaitement à notre programme de démonstration. En effet, gardons à l'esprit que l'algorithme de FFT mis en oeuvre dans IntuScope (SANDE-TOOKEY) travaille en puissance binaire. Il est donc préférable de produire des fichiers contenant un nombre de points respectant cette condition (128, 256 ...). En fait, si l'on ne tient pas compte de cette remarque, le logiciel comble les vides par des points calculés par interpolation. On peut également remplir les espaces vides avec des zéros. Il s'agit d'une autre méthode également disponible sous IntuScope. L'exemple de programmation nous a permis de stocker une sinusoïde de 100Hz, échantillonnée à 1,4kHz (**figure 6a**).

Il est évident que le 34401A ne représente pas la panacée en matière d'acquisition de données. Cependant, il représente une solution intéressante pour des périodes d'échantillonnage peu élevées. La série de courbes décrites plus bas en témoigne parfaitement.



Les lecteurs intéressés par la mise en oeuvre des équipements IEEE, se rapporteront à la série d'articles publiés par l'auteur sur le sujet et qui apparaissent en bibliographie.

LE REPLIEMENT DE SPECTRE

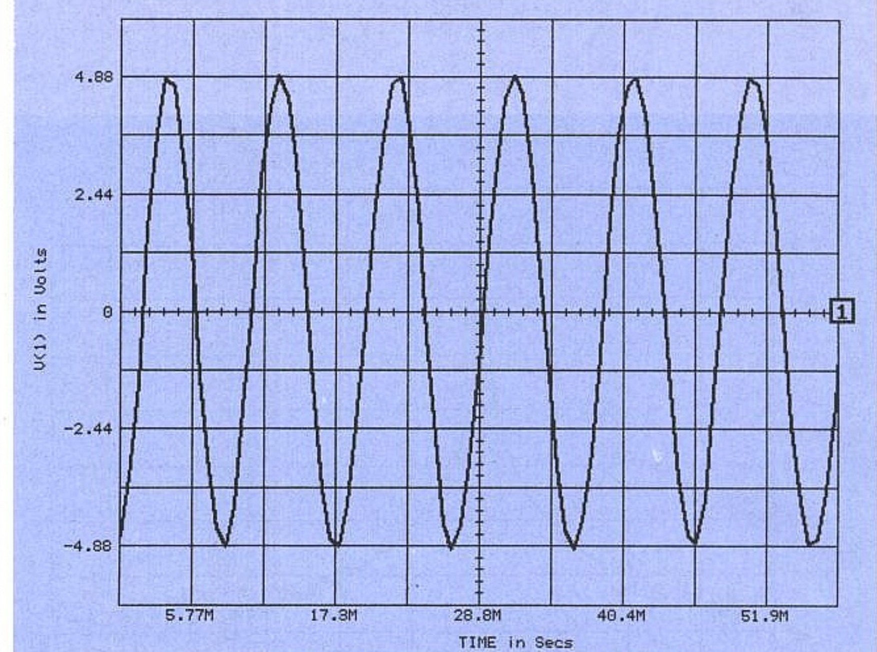
Gardons à l'esprit les problèmes liés au repliement de spectre (aliasing en anglais) et qui rendent les résultats de mesure erronés. Rappelons que l'échantillonnage d'un signal de fréquence f1 par un autre signal de fréquence f2, engendre une multitude de raies comprenant le fondamental f1 ainsi que les fréquences somme f1+f2 et différence f2–f1.

Pour illustrer ce propos, le schéma de la **figure 5a** représente un échantilloneur fonctionnant à 500 Hz auquel on applique un sinus de 100 Hz.

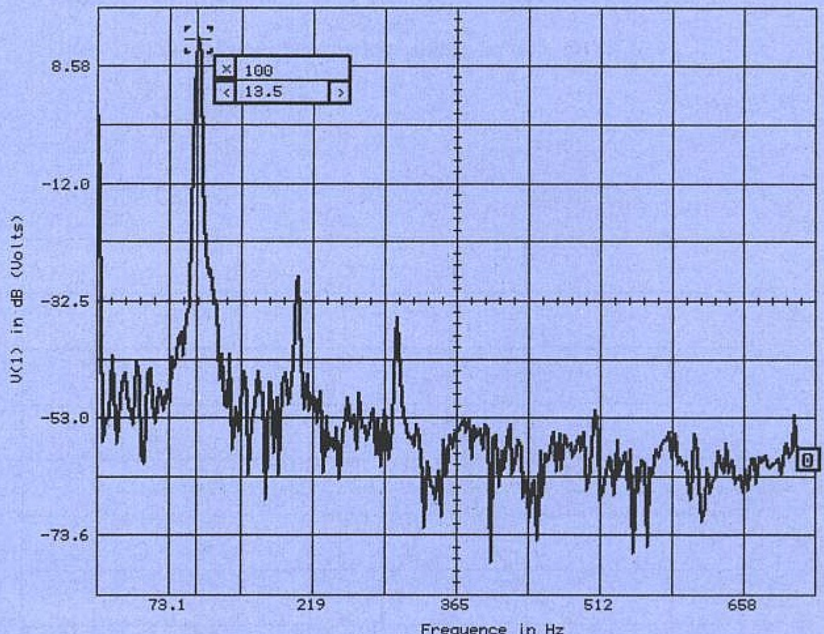
Le contenu spectral du signal de sortie simulé par IsSpice, se trouve en **figure 5b.** Le pas de stockage d'IsSpice valant 2 ms, la bande d'analyse s'arrête à 250 Hz et ne peut, par conséquent, révéler la présence des composantes somme et différence situées respectivement à 400 et 600 Hz, 900 et 1 100 Hz, etc. dont les amplitudes décroissent en sinx/x. A présent, appliquons un sinus à 400 Hz. Que se passe-t-il ? Le graphique de la **figure 5c** est éloquent. Il n'est plus possible de reconstituer le signal originel à 400 Hz mais, par contre, une raie a fait son apparition à 100 Hz. Il s'agit de la fréquence différence f2-f1 qui pénètre à l'intérieur de notre bande d'analyse. Physiquement, cette fréquence n'appartient pas au signal d'entrée mais elle se trouve bien présente dans notre fichier d'acquisition. Ce problème se retrouve sur les oscilloscopes numériques et engendre souvent des erreurs de mesure. En effet, la base de temps ajuste la fréquence de l'échantillonneur interne et, si la fréquence du signal étudié varie avec cette dernière, on se trouve en présence d'un repliement de spectre. La théorie développée autour du théorème de Shannon, indique que la fréquence d'échantillonnage doit toujours être supérieure au double de la fréquence maximale présente sur l'entrée.

En pratique, un filtre à pente raide (filtre anti-repliement, anti-aliasing filter) sera placé en entrée du système de mesure afin de bloquer toutes les fréquences susceptibles de générer du repliement de spectre. Nous illustrerons plus bas ce phénomène d'aliasing.

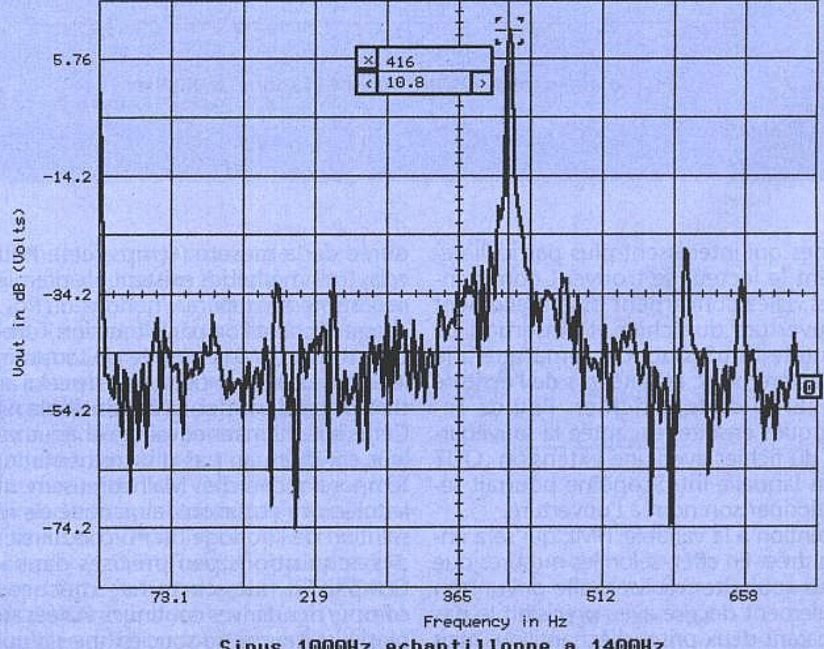
Avec le montage de la **figure 5a**, on peut s'amuser à baisser le pas de stockage d'IsSpice (à 0,5 ms par exemple) et découvrir les effets de l'interpolation mis en œuvre par le logiciel. Physiquement, le switch X4 découpe le signal d'entrée et délivre un point toutes les 2 ms. Si le pas de stockage d'IsSpice tombe à 500 µs, il faut artificiellement recréer les points absents entre deux échantillons réels. Cette méthode se nomme le suréchantillonnage (over-sampling) et trouve de nombreuses applications, notamment dans les compact-discs.



■ *Figure 6a*



■ *Figure 6b*



■ *Figure 7*

```
/*
        Programme destiné à produire une source de bruit blanc
        Christophe BASSO, Janvier 1993
*/

#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define STEP       1E-6                          /* Pas d'échantillonnage */
#define SAMPLE     1024                          /* Nombre d'échantillons */
#define path_file ".\\noi_1024.out"

void main(void)
{
   int      loop;
   FILE     *dat_file;
   float    valeur,
            temps;

   randomize();                    /* Initialise le générateur aléatoire */

/* ---------- Ouverture du fichier de données ---------------------------*/

   if ((dat_file = fopen(path_file,"wt")) == NULL ) exit (-1);

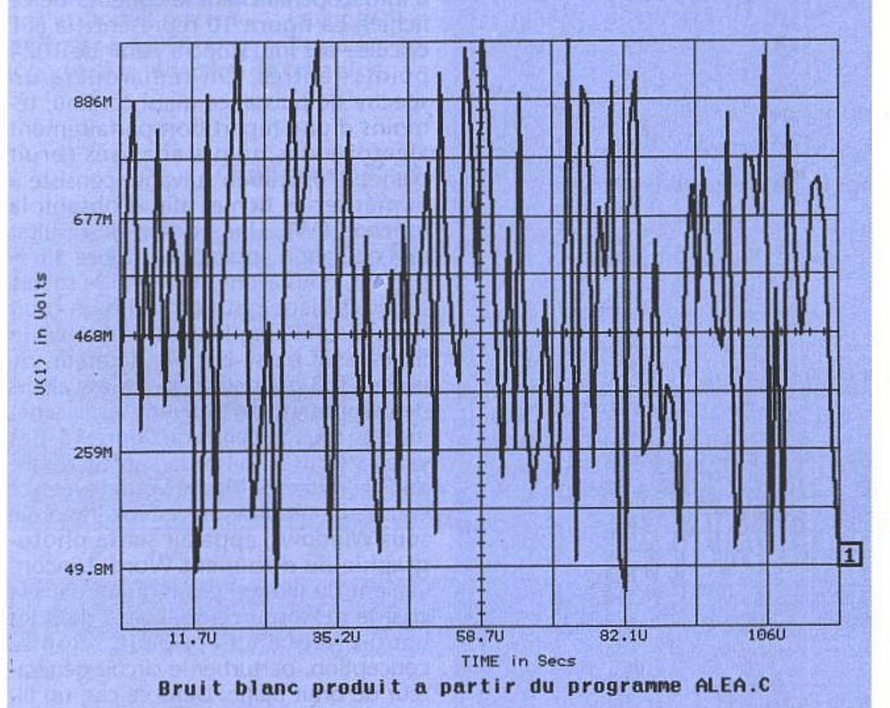
/* ---------- Ecriture de l'entete IntuScope ---------------------------- */

   fprintf (dat_file,"%s\n",".PRINT TRAN V(1)");
   fprintf (dat_file,"%s\n\n",".END ");
   fprintf (dat_file,"%s\n\n","***** TRANSIENT ANALYSIS *****");
   fprintf (dat_file,"%s\t\t%s\t\t\t%s\n\n","TIME","V(1)","INDEX");

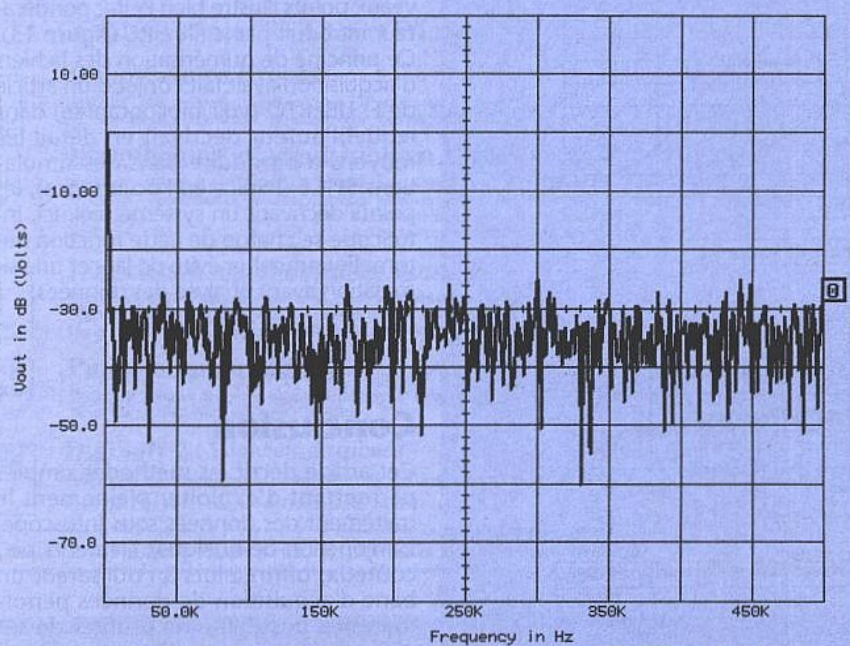
/* ------------------- Boucle de stockage de données ------------------- */

   for (loop=0; loop < SAMPLE; loop++)
   {
      temps=loop*STEP;
      valeur = (float) random(10000) / 10000 ;             /* Amplitude 1V */
      fprintf(dat_file,"%8.6f\t\t % 7.5f\t\t\t%d\n",temps,valeur,loop);
   }
      printf ("\nProgram terminated\n");
}
```

■ *Figure 8 : programme ALEA.C*



■ *Figure 9*



■ *Figure 10*

UTILISER INTUSCOPE

Le chargement du fichier de points peut se dérouler selon deux manières : sur la ligne de commande DOS, taper " scope exemple.out " suivi d'un retour chariot. Le logiciel se lance et donne ensuite la main à l'utilisateur pour travailler sur ses données. L'autre méthode revient à lancer IntuScope, puis, dans le menu " Waveforms ", ouvrir l'option " Select source data ... ". Choisir alors le fichier .OUT correspondant pour pouvoir ensuite débuter l'analyse. Par défaut, le type de fichier doit contenir une simulation TRANSIENT. Si ce n'est pas le cas, on sélectionnera le type de simulation approprié dans l'option " selected analysis ". La numérisation des courbes s'effectue simplement par la fonction "PWL". Avant de lancer cette fonction, on prendra soin de placer 0 dans l'accumulateur afin de prendre en compte la totalité des points saisis. Le résultat de l'opération se trouve dans le fichier TMP.TMP placé dans le répertoire courant. Un simple "copier-coller" (avec l'éditeur Is_Edit livré avec IsSpice par exemple) permettra une sauvegarde dans une librairie personnelle. Attention, il semble que 1024 points représente la limite du traitement d'IntuScope.

La **figure 6a** reproduit la sinusoïde 100Hz acquise à l'aide du programme de la **figure 4**, sa FFT se trouve en **figure 6b**. La présence d'harmoniques confirme la médiocrité du signal délivré par le générateur. La **figure 7** montre la FFT de ce signal dont la fréquence passe à 1000Hz, mais sous-échantillonné à 1400Hz. On remarque clairement la présence d'une raie située à 400Hz, témoins du repliement de spectre engendré par une fréquence d'échantillonnage trop faible.

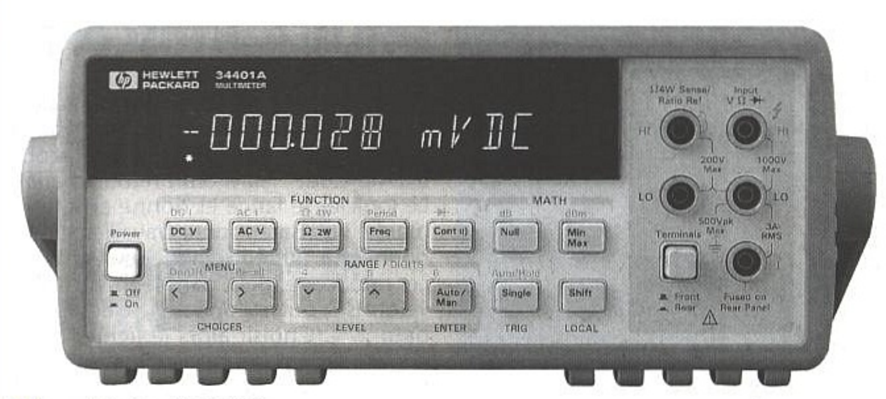
Nous aurions pu étendre l'article sur l'utilisation des FFT's avec les diverses fenêtres d'apodisation, genre Hanning ou cosinus. Un article plus complet traitant de l'apodisation et des ses effets sur les FFT's devrait paraître prochainement.

Comme nous l'exposions plus haut, IntuScope peut sauvegarder des graphiques aux formats divers : TIF, EPS, PIC, DMPL ... Cette option autorise l'import de ces images sous un traitement de texte capable de les insérer. Par exemple, Word pour windows supporte les formats TIF (Tagged Image Format) ou encore EPS (Encapsulated Postscript). Nous avons expérimenté avec succés le format TIF comme l'illustre la photographie du document ainsi réalisé.

Créer sa propre source

Grâce au fichier utilisateur, il est tout à fait possible de créer sa propre source de signal et ensuite, la numériser sous IntuScope. Le logiciel reprend en fait chaque point de l'acquisition et remplit les paramètres temps et amplitude de la fonction SPICE PWL. Par exemple, un échelon d'un volt démarrant à 10µs affecté d'un temps de montée de 100ns, s'écrira de la façon suivante : PWL (0S 0V 9.9µS 0V 10µS 1V). Le nombre de paramètres est quasi-illimité, les retours à

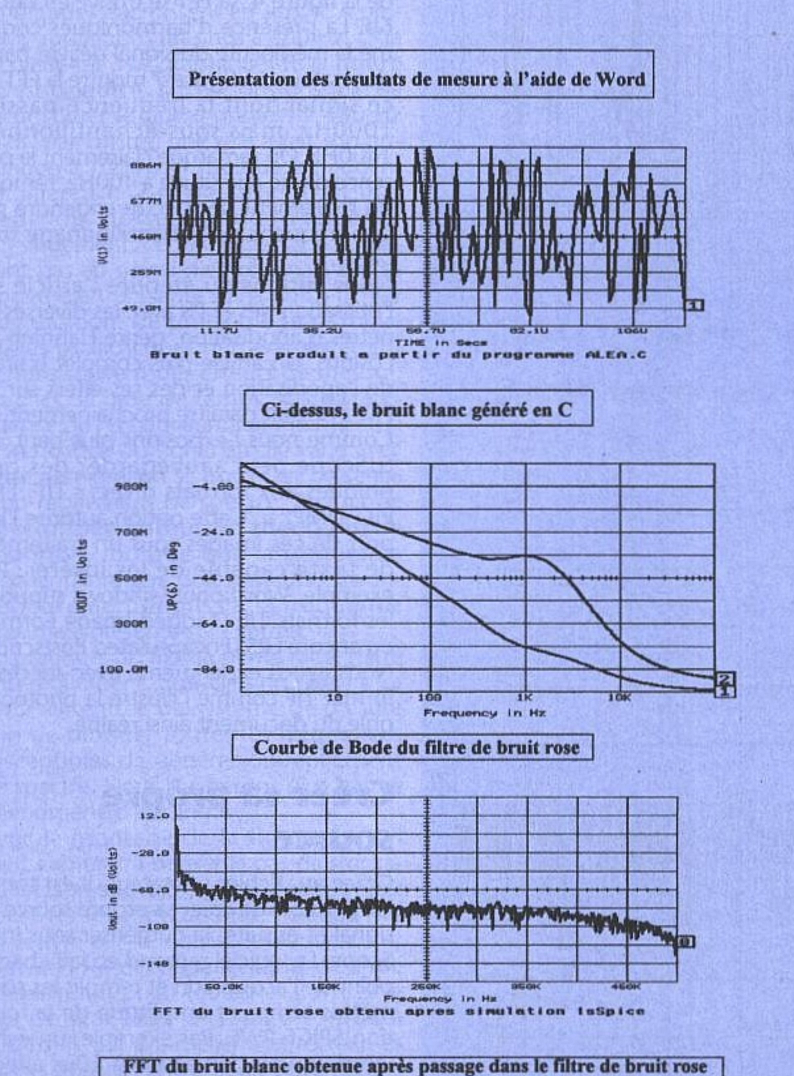




■ *Le multimètre HP 54401*

```
Piecewise Linear Data of
V(1)          PWL        0,         413M,     1.00U,     724M,
+ 2.00U,      886M,     3.00U,      299M,     4.00U,     621M,
+ 5.00U,      450M,     6.00U,      834M,     7.00U,     343M,
+ 8.00U,      657M,     9.00U,      599M,     10.00U,    386M,
+ 11.0U,      686M,     12.0U,      43.3M,    13.0U,     157M,
+ 14.0U,      895M,     15.0U,      121M,     16.0U,     462M,
+ 17.0U,      625M,     18.0U,      813M,     19.0U,     989M,
+ 20.0U,      673M,     21.0U,      141M,     22.0U,     215M,
+ 23.0U,      358M,     24.0U,      488M,     25.0U,     15.8M,
+ 26.0U,      377M,     27.0U,      787M,     28.0U,     905M,
+ 29.0U,      656M,     30.0U,      737M,     31.0U,     898M,
```

■ *Figure 11 : la fonction PWL obtenue à partir du générateur de bruit blanc.*



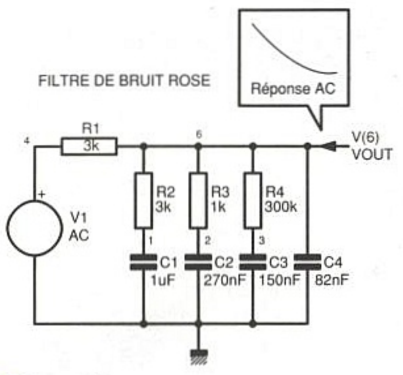
■ *Document Word format TIF*

ligne s'effectuant via le signe +. Il suffit ensuite d'introduire le résultat de la numérisation dans une librairie personnelle, et inclure les sources ainsi sauvegardées dans des simulations. On peut, par exemple, effectuer l'acquisition d'une source réelle, telle notre sinus précédent. D'autre part, l'utilisateur peut, à l'aide de quelques lignes de C, produire lui-même des résultats d'opération mathématiques sous format IntuScope et, en utilisant la fonction de numérisation, exploiter sa nouvelle source dans une simulation ultérieure. Pour illustrer nos propos, le programme de la **figure 8** produit une série de nombres aléatoires caractéristiques d'un bruit blanc. Les diverses constantes vous permettront d'ajuster le nombre de points générés, le pas d'échantillonnage et enfin, l'amplitude de la source de bruit. Un extrait du fichier résultat se trouve en **figure 3**. La **figure 9**, quant à elle, reproduit l'écran d'IntuScope affichant le contenu de ce fichier. La **figure 10** représente la FFT calculée par IntuScope à partir de 1024 points rentrés. On remarquera un spectre de puissance quasi continu, témoins d'une répartition parfaitement aléatoire des points analysés (bruit blanc). L'opération suivante consiste à numériser ce fichier afin d'obtenir la fonction PWL. Une portion du résultat de l'opération apparaît en **figure 11**. A présent, nous allons utiliser notre source de bruit blanc pour produire un bruit rose. Il suffit simplement d'ajouter un filtre passif passe-bas pondérateur, de pente -3dB par octave, dont les valeurs de composants se trouvent sur le schéma saisi sous SpiceNet en **figure 12**. Ces valeurs sont extraites de l'article publié dans le numéro 406 de votre revue. La courbe de réponse de ce filtre importée sous Windows, apparaît sur **la photographie du document Word**. L'inconvénient du filtrage passif réside dans la charge de la source qui baisse dans les hautes fréquences et peut, selon sa conception, perturber le circuit générateur de bruit blanc. Dans ce cas, un filtrage actif résout élégamment le problème (voir Radio-Plans n°459). Après simulation, la FFT lancée sur les nouveaux points illustre bien l'effet pondérateur introduit par le filtre RC (**figure 13**). Ce principe de numérisation des fichiers d'acquisition avait fait l'objet d'un article de F. UBERTO (voir bibliographie) dans lequel l'auteur décrivait en détail les moyens d'introduire dans des simulations SPICE (PSpice en l'occurrence), les points décrivant un système réel. Ici, IntuScope se charge de cette fonction automatiquement et évite de lancer une simulation avant analyse des données.

Conclusion

Cet article décrit les méthodes simples permettant d'exploiter pleinement le traitement des données sous Intuscope. L'adjonction de quelques éléments peu coûteux, offrira alors à l'utilisateur un banc d'acquisition de données performant. La possibilité de profiter de ses propres sources de signaux crées à partir d'un environnement matériel donné,

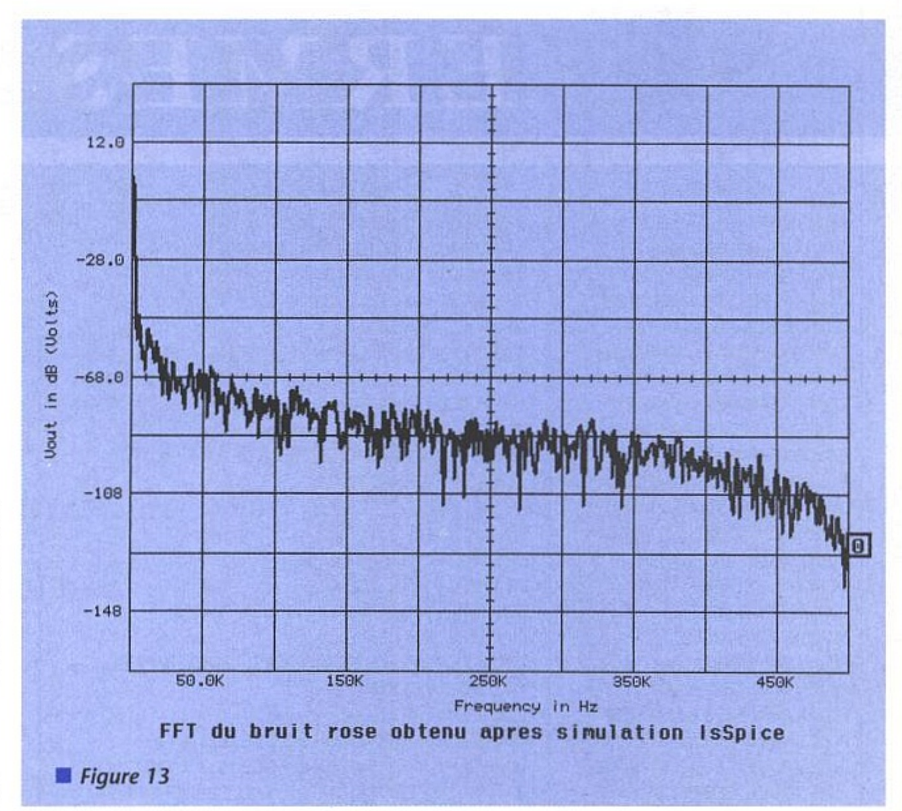




■ *Figure 12*

conduira l'utilisateur à simuler des comportements proches de la réalité. De plus, l'enrichissement apporté par l'importation de courbes dans les rapports d'analyse contribuera à valoriser davantage le produit présenté. L'auteur tient à remercier F. UBERTO pour l'aide qu'il a apportée à la rédaction de cet article.

Christophe BASSO



■ *Figure 13*

Bibliographie

Sur IsSpice et la simulation de données réelles :

Simuler des données réelles avec PSpice, F. UBERTO, Electronique Radio-Plans n°534

Le simulateur SPICE d'Intusoft, C. DUCROS, Electronique Radio-Plans n°543

Sur l'utilisation du bus IEEE-488 :

La programmation du HP34401A, C. BASSO, Electronique Radio-Plans n°536

Le voltmètre HP34401A, C. BASSO, Electronique Radio-Plans n° 535

Introduction au langage SCPI, C. BASSO, Electronique Radio-Plans n°537

La programmation des cartes IEEE pour PC, C. BASSO, Electronique Radio-Plans n°536

Introduction au bus IEEE-488, C. BASSO, Electronique Radio-Plans n°534

La norme IEEE-488.2, C. BASSO, Electronique Radio-Plans n°537

INTERFACE DE PUISSANCE POUR MOTEUR C.C.

Cette interface de puissance permet la commande de moteur à courant continu de forte intensité. Elle est bien adaptée au contrôle de vitesse et peut s'intégrer dans une boucle d'asservissement de position. De plus, elle peut s'adapter à des commandes de différentes provenances.



Une interface de puissance électromécanique pose différentes contraintes, notamment une vitesse de commutation faible. Pour un contrôle de vitesse par exemple, seule une carte électronique sera adaptée. Parmis les composants électroniques spécialisés, certains, comme le NE544 utilisé en radiocommande sont réservés à des applications de faible puissance. D'autres (ex:UDN2936), limités à quelques ampères sont d'approvisionnement parfois difficile.

L'interface décrite ici a les caractéristiques suivantes :

- entrées compatibles TTL, CMOS, microcontrôleur et NE544,
- commutation électronique,
- courants de sortie maximum : 15A (réversible),
- freinage électrique optionnel (dipswitch),
- très faibles pertes calorifiques (0,15Ω de Rds on),
- entrée inhibition (découpage, surveillance courant...),
- protection contre les baisses de tension,
- alimentation 12V,
- courant de repos très faible (moins de 20mA),
- dimensions réduites (115 x 90 mm),
- composants standards,
- coût faible.

La commande du moteur se fait à l'aide d'un pont de MOSFETs, piloté par quatre AOP, ce qui évite les drivers spécialisés. Leur commande est assurée par des circuits logiques classiques. La carte dispose de trois entrées logiques : X, Y et INH. X et Y déterminent le sens de rotation, alors que INH sert d'inhibition. De plus, un dipswitch SW1 permet de mettre en/hors service le freinage électrique. Seule l'entrée INH est en logique inverse, comme le montre le **tableau 1** :

L'état (X,Y)=(1,1) provoque un freinage électrique quelle que soit la position de SW1. Si la commande est assurée par un circuit de radio-commande (ex:NE544), cet état n'existe pas. Le freinage électrique reste donc, pour ce type de commande, une option. Si la commande est faite par un circuit numérique (ex:microcontrôleur), le dipswitch SW1 reste volontairement ouvert. En effet, les quatre états sont alors disponibles : hors service (0,0), rotation normale (1,0), rotation inverse (0,1) et freinage électrique (1,1).

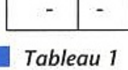
Les entrées X,Y et INH sont dirigées vers un circuit logique, 74HCT08. Le choix de cette technologie permet d'être compatible aussi bien avec des signaux TTL ou CMOS, qu'un microcontrôleur, ou encore les sorties d'un NE544. Il faut toutefois que ces signaux n'excèdent pas 5V et qu'ils soient référencés par rapport à la masse. Si les entrées ne sont pas connectées, l'interface travaille en sécurité : X, Y et INH sont au niveau bas forçant les transistors T1, T2, T3 et T4 au blocage.

La porte U1A (INH et SECU) est normalement à l'état haut. Le passage à l'état bas de l'un des signaux provoque le passage à l'état bas de U1A, et par conséquent le passage à l'état bas des portes U1B, U1C, U1D, respectivement associées aux transistors T1, T2 et T3. Les trois transistors ainsi bloqués, il n'y a plus de tension aux bornes de MOT+,MOT- ni même de freinage électrique.

Le schéma de l'interface de puissance est donné **figure 1**.

Chaque transistor est commandé par un AOP LM358 utilisé en comparateur. L'alimentation de ce circuit fournit la tension de commande du transistor. Pour bien saturer les MOSFETs, il faut appliquer sur la grille une tension supérieure de plus de 10V à celle présen-

X	Y	F	INH	T1	T2	T3	T4	ETAT
0	0	0	1	0	0	0	0	HORS SERVICE
1	0	0	1	1	0	0	1	SENS NORMAL
0	1	0	1	0	1	1	0	SENS INVERSE
1	1	0	1	1	0	1	0	FREINAGE ELECTRIQUE
0	0	1	1	0	1	0	1	FREINAGE ELECTRIQUE
1	0	1	1	1	0	0	1	SENS NORMAL
0	1	1	1	0	1	1	0	SENS INVERSE
1	1	1	1	1	0	1	0	FREINAGE ELECTRIQUE
-	-	-	0	0	0	0	-	HORS SERVICE


Tableau 1



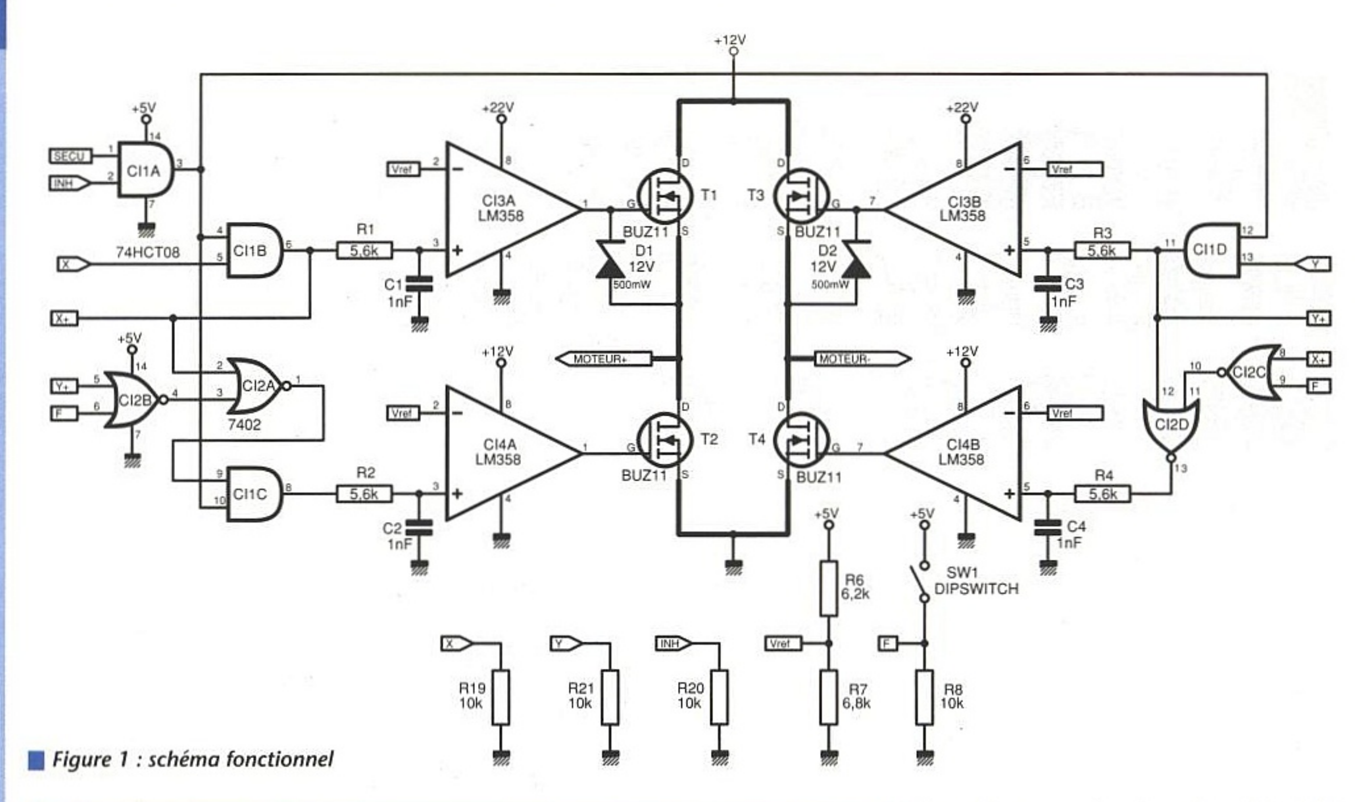
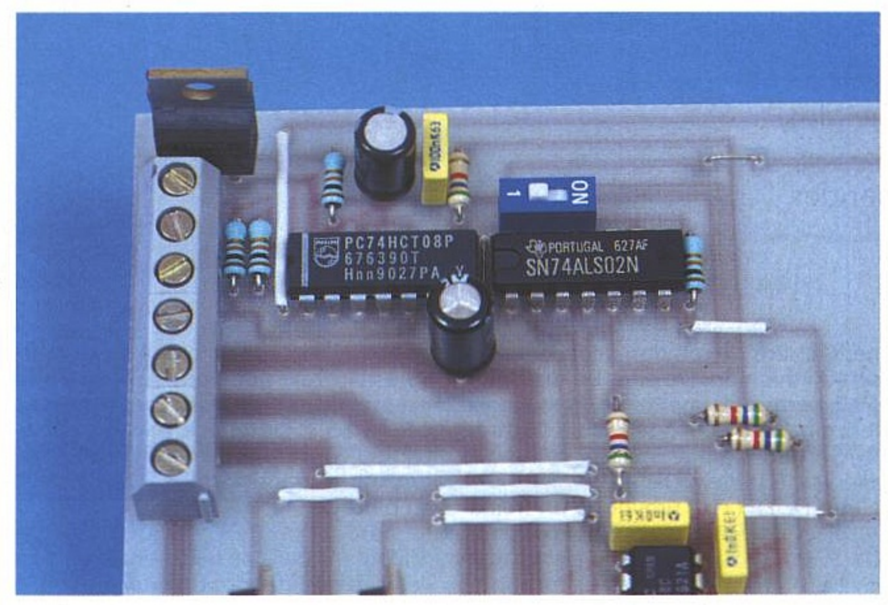


Figure 1 : schéma fonctionnel



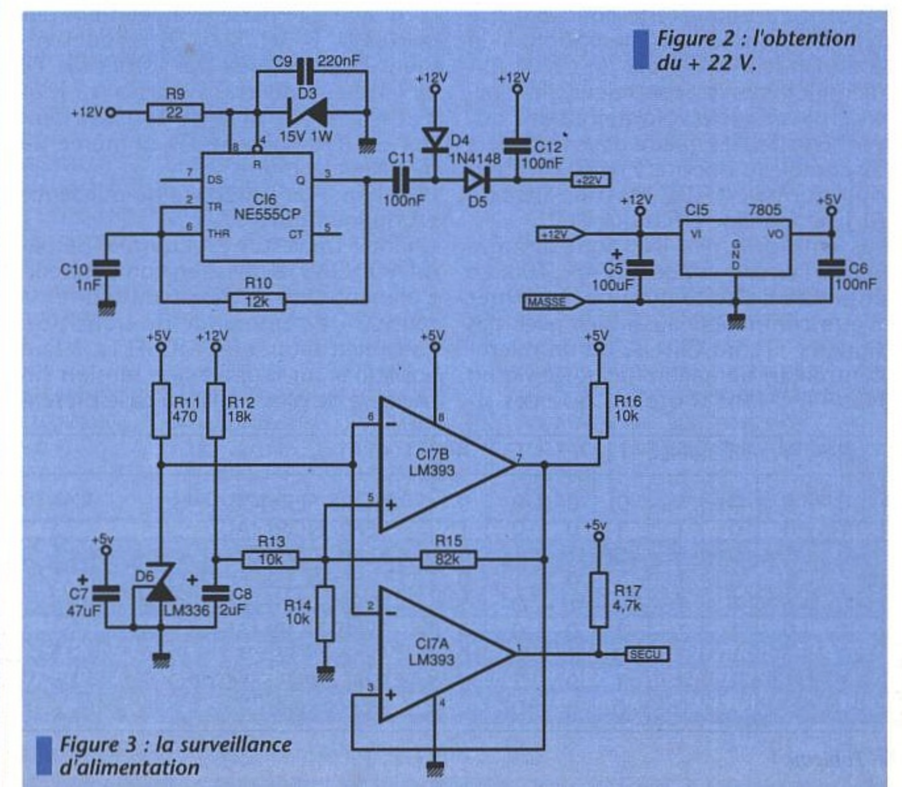


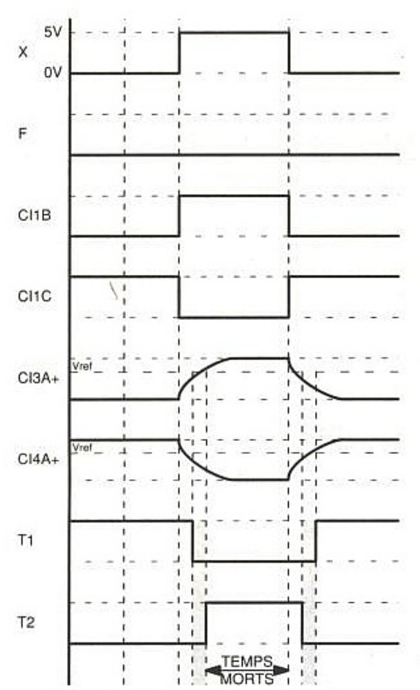
Figure 2 : l'obtention du + 22 V.

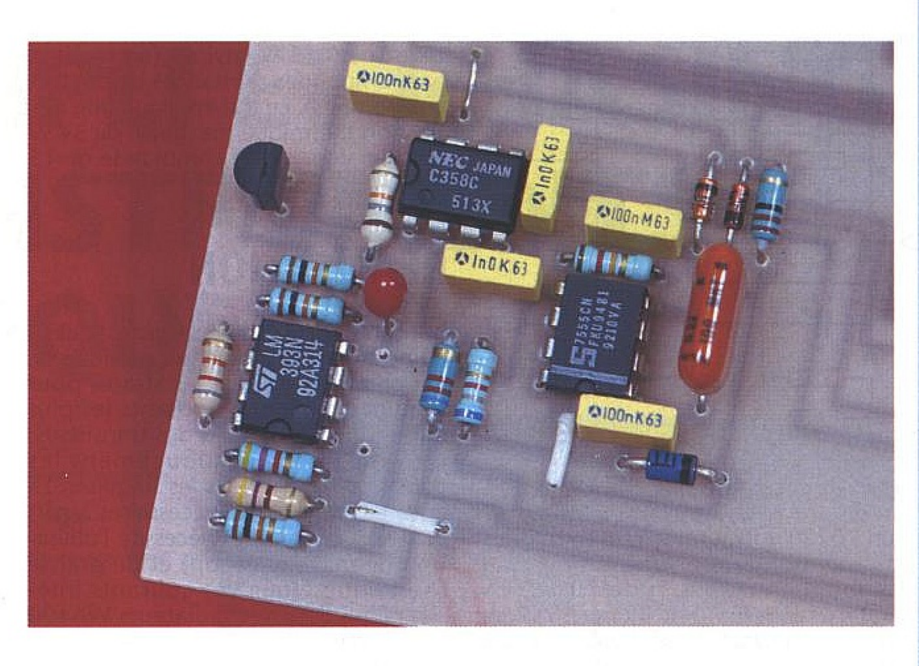
Figure 3 : la surveillance d'alimentation

te sur la source. Les transistors T2 et T4 étant à la masse, la commande (l'alimentation de U4), est +12V. Par contre, pour les transistors T1 et T3, il faut une tension de commande de 12 + 10 = 22V (alimentation de U3). Les diodes D1 et D2 veillent à ce que cette tension ne soit pas supérieure de plus de 12 volts par rapport à la source.

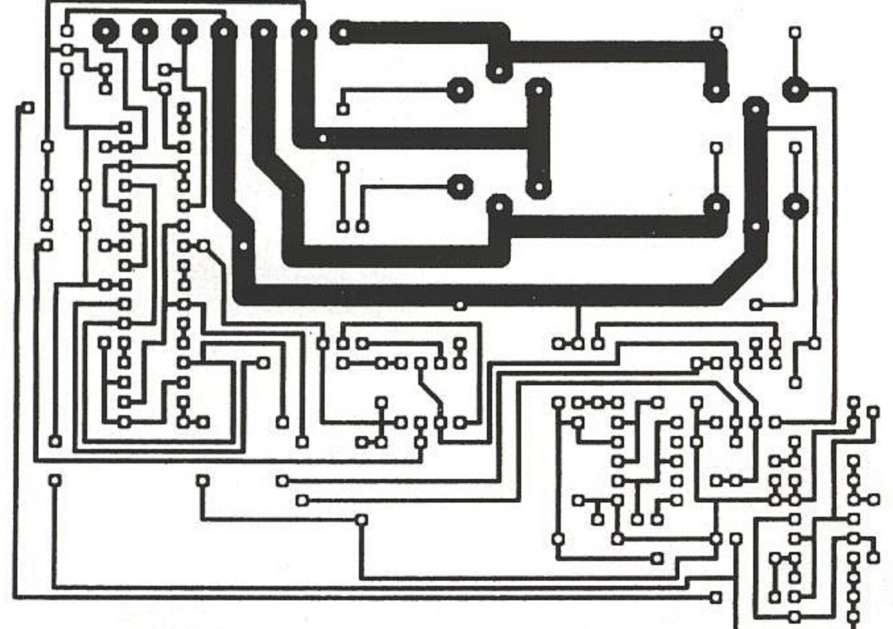
On fabrique le +22V (une tension flottante de 10V référencée par rapport au +12), par un petit doubleur à pompe à diodes (**figure 2**). Le circuit U6 assure l'oscillation nécessaire. Lorsque la sortie Q est au niveau bas, le condensateur C11 se charge à 12-0,7=11,3V par D4. Lorsque la sortie Q est au niveau haut, le potentiel de C11 monte à 11,3+12=23,3V. Le condensateur C12 se charge alors à travers D5, ce qui donne une tension de sortie de 23,3-0,7=22,5V. Pour cela, il est nécessaire que le niveau bas de Q soit très proche du zéro, et son niveau haut très proche de 12V. Seule l'utilisation d'un 555 de technologie CMOS permet d'obtenir ces niveaux(NE555CP). Un NE555 normal ne fournit de 19V, au risque d'avoir une saturation trop faible, et de faire chauffer les transistors lors de la commande de forts courants.

De même, si le +12V vient à chuter, les transistors risquent de ne plus être suffisamment saturés. Le module de SURVEILLANCE D'ALIMENTATION (**figure 3**) inhibe, par le passage à l'état bas de SECU, la commande du pont pour une tension d'alimentation inférieure à 8V, et ne rétablit le fonctionnement que si celle-ci est remontée à 9,5V. Cette surveillance est réalisée par un comparateur LM393. Celui-ci compare la branche R11,D6 qui sert de référence de tension, à la branche R12,R13,R14. Le condensateur C8 permet de gommer les micro-chutes de tensions. La résistance R15 provoque l'hystérésis nécessaire. La sortie du premier comparateur suffirait, mais le boîtier en comportant deux, le second comparateur permet d'avoir une transition bien nette. Le LM393 est un double comparateur à sortie collecteur ouvert, par conséquent il faut mettre une résistan-

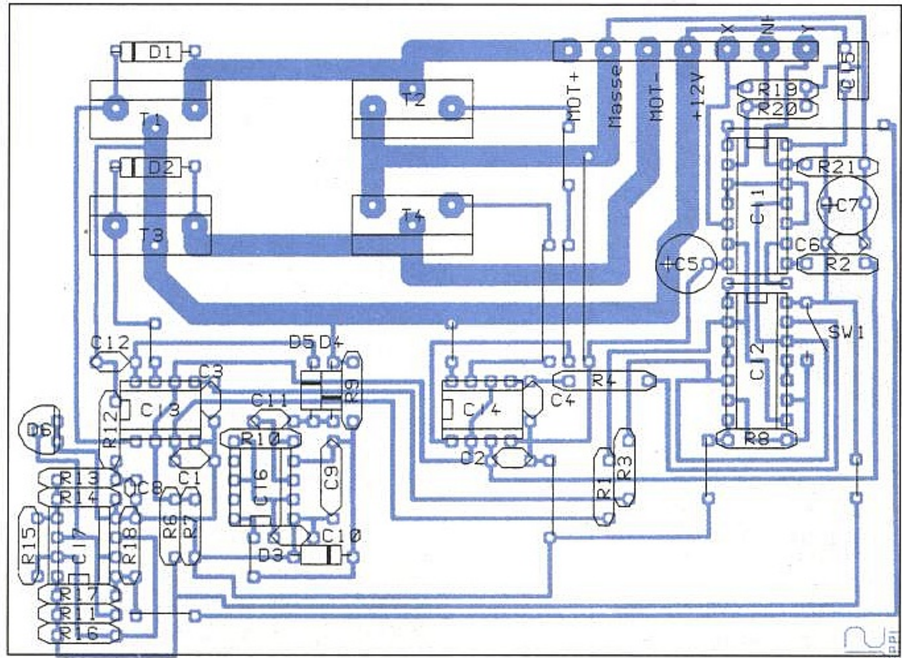




■ *Figure 4 : chronogramme de commande*



■ *Figure 5 : le circuit imprimé*



■ *Figure 6 : l'implantation correspondante*

ce de pull-up sur la sortie SECU (R17). Dans un montage en pont, une des priorité consiste à vérifier que les deux transistors d'une branche ne conduisent pas simultanément, sans quoi la carte est confrontée à un léger problème: 0,15Ω sous 12V, nous vous laissons calculer...

Afin d'y remédier, on s'assure tout d'abord que la logique de commande ne fournisse jamais ce cas :

* si X=1, alors U1B=1 —> U2A=0 —> U1C=0 —> T2 BLOQUE
* si X=0, alors U1B=0 —> T1 BLOQUE

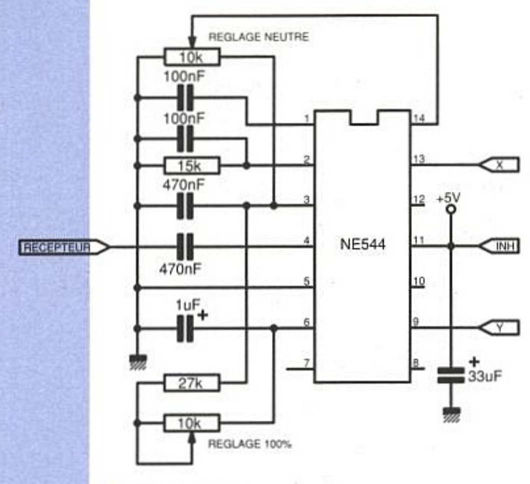
Quel que soit l'état de X, il y a toujours un transistor de la branche bloqué, par conséquent jamais de court-circuit. Pour la branche T3,T4, le raisonnement est analogue.

Le second problème est lié aux régimes transitoires (cf C.BASSO). Que ce passe-t-il lors de l'inversion du sens de rotation, les transistors passants devenant bloqués et inversement. Pour bien comprendre, il faut analyser le chronogramme (**figure 4**). Le temps de retard provoqué par la charge de RnCn et le seuil de commutation Vref, interdit tout court-circuit de commutation (les retards de commutation des transistors sont de l'ordre du dixième de microseconde). La modification du seuil de tension Vref augmente ou diminue l'écart entre la coupure d'un transistor et la conduction de l'autre. Une valeur trop basse crée une chevauchement des temps de conduction. Sans ces temps morts, le courant de court-cricuit monterait à une centaine d'ampères pendant quelques dixièmes de microsecondes.

Des courants de 15 A nécessitent un refroidissement des transistors. Le choix du radiateur se fait en fonction de : P la puisance dissipée, Tj la température de jonction maximale, Ta la température ambiante, Rjb la résistance thermique entre jonction et boîtier, Rbr entre boîtier et radiateur, et Rra entre radiateur et air. La relation est : Rra = (Tj - Ta) / P - (Rjb + Rbr).

Le constructeur donne pour un BUZ11 : Tj = 175° C, Rjb = 1,67° C/W et Rds = 0,04 Ω. De plus, Rbr = 0,5° C/W pour





■ *Figure 7 : adaptation au NE 544 (RC)*

un T0220 à fixation par vis avec graisse thermique. Pour une température ambiante maximum de 70° C, P = 0,04 x 15^2 = 9 W donc Rra = 9,5° C/W. La valeur constructeur standard la plus proche est 9° C/W, ce qui ne posera pas de problème, l'efficacité du radiateur étant inversement proportionnelle à sa résistance thermique. Un modèle WA200 semble bien adapté.
Les pertes calorifiques de la carte sont dues d'une part à la résistance des jonctions source/drain des transistors (0,04 Ω) et d'autre part à la résistance des pistes. Quel que soit le sens de fonctionnement, le courant passe par deux transistors, par conséquent, il y a 0,08 Ω en série avec le moteur. La carte a une résistance de piste (partie puissance) de 0,06 Ω. Par conséquent, la résistance totale du module puissance est de 0,15 Ω environ (on ne tient pas compte des résistances de contact aux borniers), ce qui implique une dissipation thermique maximale de 34 W pour un courant de commande de 15 A.

REALISATION

Le montage de la carte (**figures 5 et 6**) ne pose pas de problèmes particuliers. On commencera par les straps, pour terminer par les composants les plus importants. Au niveau des transistors, on veillera à plier doucement les pattes, afin de ne pas les fragiliser. La pose des radiateurs (nécessaires à partir de 4A de charge) nécessite l'utilisation de mica (isolation) et de graisse thermique. Pour des courants inférieurs à 8,5 A, des radiateurs WA400 sont suffisants. L'utilisation de WA200 sans graisse thermique, réduit l'intensité maximale à 12,5 A.
Le choix d'un boîtier avec ouïes d'aération, de préférence métallique, sera conseillé (ex : ESM AT-13). Si le montage est placé dans un environnement soumis aux vibrations (automobile), les entretoises seront de préférence en nylon à fixation par clip.
Cette interface a été utilisée pour commander un moteur à courant continu de 5A (moteur Bosch issu d'un essuie-glace). En utilisation automobile, on veillera à isoler les bornes + et - de la masse. C'est en général déjà le cas sur les moteurs du type «vitres électriques», mais ce n'est pas celui des moteurs d'essuie-glace. Il suffit d'ouvrir le cache-moteur et de couper la connexion reliant une borne au boîtier.
Les amateurs de radiocommande désireux de faire une commande de vitesse pourront utiliser cette interface avec un circuit NE544 (**figure 7**).

Serge LANDERRETCHE

Bibliographie

Memotech, Collection A. Capliez, Editions CASTEILLA (EL Educative)
Electronique Radio-Plans, N° 506, article de C. Basso p. 67.



NOMENCLATURE

Résistances

R1, R2, R3, R4 : 5,6 kΩ 1/4W
R6 : 6,2 kΩ 1/4W
R7 : 6,8 kΩ 1/4W
R8 : 10 kΩ 1/4W
R9 : 22 Ω 1/2W
R10 : 12 kΩ 1/4W
R11 : 470 Ω 1/4W
R12 : 18 kΩ 1/4W
R13, R14 : 10 kΩ 1/4W
R15 : 82 kΩ 1/4W
R16 : 10 kΩ 1/4W
R17 : 4,7 kΩ 1/4W
R19, R20, R21 : 10 kΩ 1/4W

Condensateurs

C1, C2, C3, C4 : 1 nF MKT 63 V
C5 : 100 µF 16 V radial
C6 : 100 nF MKT 63 V
C7 : 47 µF 16 V radial
C8 : 2,2 µF tant. goutte
C9 : 220 nF MKT 63 V
C10 : 1 nF MKT 63 V
C11, C12 : 100 nF MKT 63 V

Semi-conducteurs

T1, T2, T3, T4 : BUZ11 ou BUZ11A
U1 : 74HCT08 (pas de 74HC08)
U2 : 74LS02 ou similaire
U3, U4 : LM358N boîtier DIL08
U5 : 78L05 ou régulateur 5V
U6 : NE555CP ou 555 CMOS
U7 : LM393N
D1, D2 : 12V 0,5W
D3 : 15V 1,3W
D4, D5 : 1N4148
D6 : LM336

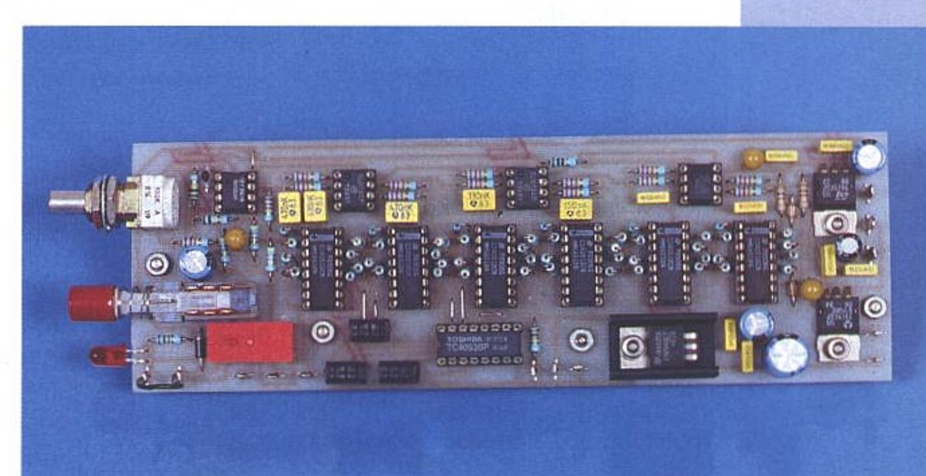
Divers

Radiateurs WA200 percés (x4)
Intercalaires isolants mylar pour TO220 (x4)
Epoxy simple face cuivré 35µm 115x85mm (x1)
Connecteur bas profil type standard (Camden) deux points (x2)
Connecteur bas profil type standard (Camden) trois points (x1)
SW1 : dipswitch 1 inter.



PRÉAMPLI ET PHASING POUR INSTRUMENTS DE MUSIQUE

L'étude que nous proposons ici intéressera plus particulièrement les lecteurs jouant d'un instrument traditionnellement (mais improprement) qualifié d'«électrique».



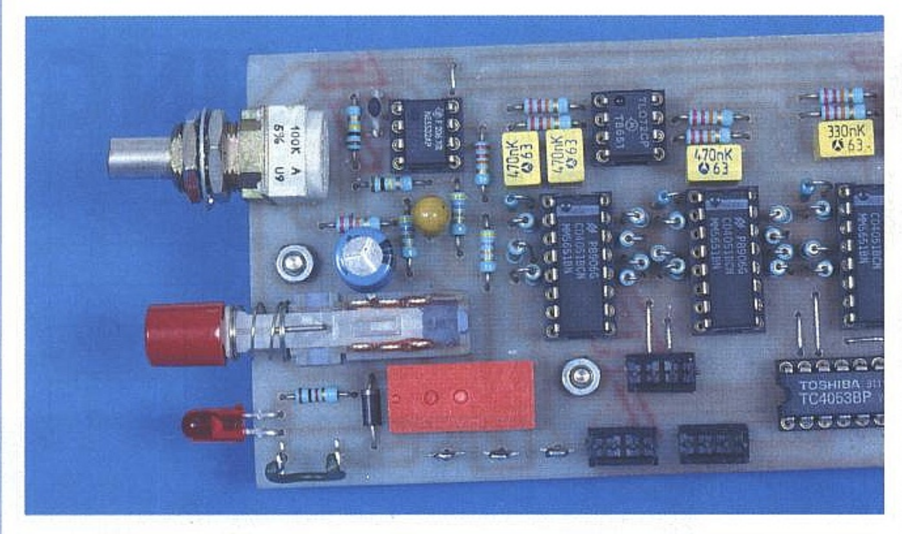
Il est bien rare, en effet, que la chaîne d'amplification de tels instruments soit linéaire, et, souvent, de nombreux générateurs d'effets viennent encore s'insérer, comme le phasing que nous allons voir cette fois. Ce dernier présentera la particularité de pouvoir être commandé de diverses manières, notamment par PC, bus I2C ou logique classique (manuelle ou automatique) ; mais au préalable, nous examinerons en détails un petit module servant d'étage d'entrée et qui malgré son extrême simplicité comporte de nombreux avantages.



PRÉAMPLI INGUIT

Qui n'a jamais tenté de raccorder une guitare «électrique» à une chaîne HI-FI, sans trouver LA solution idéale ?
Des entrées AUX à trop faible gain, aux entrées micro (quand elles existent) saturant bien vite ou dénaturant le son, il est quasiment impossible d'arriver à un compromis satisfaisant. Rien d'étonnant à cela, car les capteurs ont souvent des caractéristiques particulières qui sortent des offres d'entrées classiques d'une installation audio domestique.
Nous ne parlerons pas des enceintes qui n'ont rien de commun avec les «HPs» des amplis de guitare, et qu'il serait dangereux de solliciter comme tels. Tout au plus admettrons-nous de mélanger des modulations «instruments» avec celles d'un laser ou d'une cassette et d'écouter le tout au casque dans de bonnes conditions, voire enregistrer.
Notre but n'est pas pour cette fois de décrire une «Sound Machine» prête à l'emploi, mais de donner plus modestement quelques idées pour compléter une installation HI-FI et surtout réfléchir aux problèmes particuliers posés par les capteurs d'instruments.
Ces derniers présentent souvent une impédance élevée (20 à 100 kΩ) et une tension de sortie très variable. En pratique, un accord délié doucement peut fournir 25 mV alors que si on le plaque avec une certaine violence, 250 mV sont fréquents. Si on admet 20 dB (c'est un minimum) pour laisser «fondre» une note, on arrive à 40 dB de dynamique. Les niveaux se situent donc entre «micro fort» et «petite ligne» haute impédance, ce qui explique les difficultés de raccordements déjà citées.
Mais le plus gros du problème est essentiellement dû à la haute impédance et à la non-symétrie des liaisons, ce qui a pour conséquence de récupérer mille bruits parasites et de les transmettre aux étages d'entrées ; sans oublier la forte sensibilité des capteurs aux rayonnements électromagnétiques tels que ceux produits par les transformateurs d'alimentation.
La solution que nous proposons (visible **figure 1**) permet d'obtenir un gain global de 22 dB, la saturation étant alors mesurée en sortie chargée sous 600 Ohms à 4,8V soit +16 dBU.
Le lecteur attentif aura remarqué une perte de 8 dB dans la chaîne: en effet, le premier étage (IC1a) au gain de 10 donne 20 dB et le second (IC1b) ajoute





10 dB. Les 8 dB «perdus» le sont dans le diviseur constitué de R Photo-résistance de IC2 (que nous appellerons R-ph pour aller vite) et R4 .

Il est facile de déterminer par calcul la valeur réelle de R-ph. Quand la LED de IC2 est allumée : environ 6,8 kΩ ; lorsqu'elle est au repos, on mesure une atténuation de 50 dB, soit R-ph = 1,4 MΩ, toujours pour R4 = 4,7 kΩ.

Toutefois ces performances sont loin de celles garanties par le constructeur des photo-résistances, lequel annonce 10MΩ/4,7kΩ.

Il faut dire que R-ph montée sur la maquette de l'auteur est un élément provenant d'un lot de 10 pièces à prix très avantageux, ce qui pourrait expliquer le «hors-normes».

Pourtant les résultats sont très acceptables dans les pires conditions, puisqu'au mieux on serait en mesure d'espérer une perte de 6 dB et un affaiblissement de 65, et nous obtenons avec un composant de «seconde source» respectivement -8 et -50 !

Le simple fait de porter la cathode de la LED contenue dans IC2 à 0V, permet donc de quasiment fermer la voie (gate), mais de nombreuses autres commandes sont également envisageables. En bas de la figure 1, nous en proposons une toute simple puisqu'elle n'exige aucune électronique complémentaire. Il s'agit simplement de permettre à un câble d'être «en l'air» et d'interdire à l'ampli de considérer le 50 Hz comme une modulation à diffuser plein pot ...

La méthode ne nécessite que quelques précautions, dont la plus importante est de s'assurer (peu fréquent) que la guitare est protégée du continu. Un condensateur en série dans la ligne n'est pas un luxe et 100 nF 100V conviendront très bien.

Cette manipulation pourra être faite à l'occasion d'un changement de cordes, c'est à dire quand la guitare peut être démontée aisément.

On en profitera pour remplacer le jack châssis d'origine par un modèle stéréo câblé mono... En effet, les avantages sont importants: meilleur guidage mécanique de la pige du jack mâle (trois points au lieu de 2) et compatibilité totale avec les entrées traditionnelles, dont INGUIT.

En remplaçant le câble 2 fils classique par une paire blindée, rien ne changera pour une utilisation sur des amplis du commerce mais avec INGUIT, un arrachage du câble côté guitare ou côté ampli conduira à une fermeture de la voie de 50 dB minimum : MUTE n'est plus au 0V. Ceci n'est qu'un exemple et nous en découvrirons d'autres, mais au préalable, il serait bon de donner quelques précisions sur IC2.

La **figure 2** en donne la construction. Ce n'est pas - comme on aurait pu le penser - un circuit «du commerce", mais un assemblage maison effectué sur un support 14 broches tulipes, dont seuls les quatre points utiles ont été gardés, soit : 7 et 14 pour la photo-résistance , 1 pour l'anode de la LED avec résistance de limitation et 8 pour la cathode.

Le tout est ensuite placé dans un morceau de gaine thermo-rétractable noire, puis noyé dans l'Araldite.

C'est une formule comme une autre, l'essentiel étant de placer la photo-résistance dans des conditions optimales d'obscurité pour bénéficier d'une atténuation maximale.

La maquette a été enfermée ensuite dans un boîtier métal clos, ce qui a eu pour double effet d'éviter la «ronflette", mais aussi d'interdire à toute source de lumière extérieure -autre que la LED intégrée- d'atteindre et donc de modifier R-ph. A titre indicatif, un spot de 200W placé à 20 cm de l'ensemble, n'a pas modifié d'un seul dB les mesures effectuées préalablement dans la quasi obscurité.

La **figure 3** propose quelques idées d'exploitations de IC2.

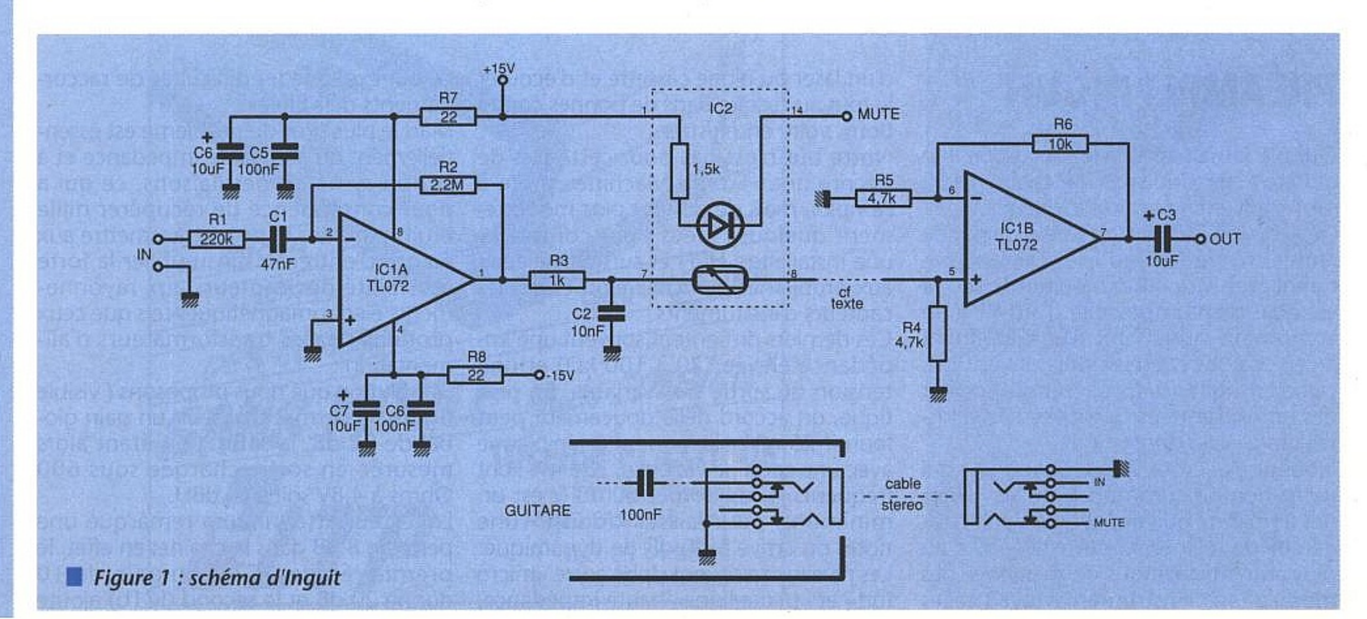
Outre une commande MUTE (comme celle que nous avons vue mais qui pourrait aussi être conjointement - voire uniquement - activée par une pédale extérieure), il serait aisé de construire un «noise-gate", en amplifiant puis redressant le signal post-mute, et en le comparant à un seuil minimum acceptable, en vue d'ouvrir ou de fermer la porte constituée d'une seconde section IC2.

Ceci est simplifié, et chacun aura compris qu'une seule et même cellule R-ph peut servir aux trois fonctions citées, au prix d'un minimum de logique additionnelle.

Nous n'en avons pas parlé, mais une particularité du système est que les ouvertures et fermetures de la porte se font en douceur; avec une ouverture relativement rapide (20 à 50 ms) et une fermeture lente (4 secondes pour -50 dB dues à la saturation de R-ph après éclairement de la LED) aucun condensateur n'étant alors en cause.

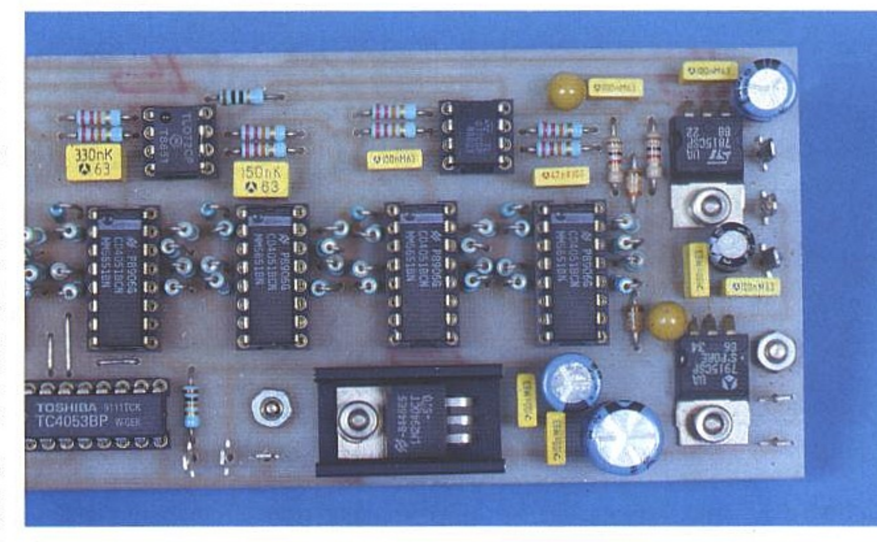
Tout ceci est parfait en audio, surtout dans CE sens : il ne serait pas très raisonnable de vouloir commander R-ph par une LED pour une fonction limiteur ou compresseur.

Au besoin - mais après de bons calculs et un judicieux choix dans l'emploi de cette (ou ces) résistance(s) variable(s)



■ *Figure 1 : schéma d'Inguit*

afin de surveiller la bande passante - une étude très intéressante pourrait être menée combinant LED et ampoule à incandescence, en vue d'obtenir des constantes de temps satisfaisantes pour des «effets» garantis silencieux.
La figure 3 propose encore une solution: le «vibrato». Il est vrai que cet effet est provisoirement passé de mode. Pourtant bien utilisé - c'est à dire un bon choix des capteurs actifs (et de leurs phases) sur une guitare dont la lutherie élabore des «delays» naturels -, avec très peu de moyens et surtout sans traitement particulier, on obtient des résultats surprenants.
Un simple signal de forme triangulaire et à basse fréquence variable pour piloter la LED, plus un potentiomètre placé en parallèle avec R-ph pour doser l'amplitude de l'effet, et voici un excellent vibrato.
Pour ceux qui ne le sauraient pas, l'effet consiste en une modulation d'amplitude que l'instrumentiste doit se charger de synchroniser avec les cycles de l'horloge.
Pour «voir», on pourra commencer par commander la LED de IC2 par un générateur de fonctions, que l'on remplacera ensuite par un montage très simple dont la fréquence sera ajustable de 0,5 à 25 Hz environ.
Dernières remarques et variances autour de INGUIT :
Le fait de placer une porte de bruit en tout début de chaîne présente de nombreux avantages, comme par exemple qu'elle n'intervienne pas sur les effets placés APRES. Ainsi, une réverbération pourra continuer à sonner malgré la fermeture de la porte.
On conçoit que c'est très important , car une porte qui serait insérée au final devrait alors avoir son seuil réglé pour de très faibles niveaux, d'où une efficacité considérablement amoindrie.
Par contre, des inserts du genre transposeurs d'octaves (pratiqués généralement staccato) ne pourront pas être traités correctement par INGUIT : il faudrait une fermeture de porte bien plus rapide (et parfaitement réglée) pour bénéficier des avantages de ce type d'effet, sans en subir les inconvénients bien connus des utilisateurs.
Enfin, est-il utile d'indiquer que ce montage précédé d'un transformateur du type SD41B de MILLERIOUX, constituerait un très bon préampli micro, basse impédance et symétrique ?
L'auteur pense plus particulièrement à la sonorisation des lieux de cultes et des



salles de conférences, pour lesquels une astucieuse exploitation du MUTE ne serait souvent pas un luxe !
Il n'y a en général aucun opérateur pour veiller les réglages, et on peut être surpris de constater - au moyen d'un casque branché sur la modulation diffusée effectivement par les haut-parleurs - la présence de signaux sans intérêt: est-il bien nécessaire par exemple d'augmenter encore la réverbération naturelle d'une église quand les grandes orgues jouent, en rediffusant dans des enceintes ordinaires ce que des micros ordinaires «récupèrent» 30 mètres plus loin ?
Dans ce cas précis, de nombreuses astuces simples et peu coûteuses devraient être mises en oeuvres, comme des capteurs de charges devant les pupitres ou dans la chaire, activant une entrée MUTE du type INGUIT (voire des barrières IR).
Contrairement à ce qu'on pourrait supposer, une détection de présence sur le « poste de travail » est alors bien plus efficace qu'un noise gate, même de très grande qualité et réglée avec soin : il est matériellement impossible de trouver un compromis raisonnable pour sélectionner uniquement à partir d'une détection de niveaux la parole, des grandes orgues « pleins pots ».

Réalisation

Pour construire INGUIT, le circuit imprimé livré **figure 4** suffit, et il est conçu pour être enfermé dans un boîtier HF TEKO ref 371 (53 x 50 x 26).
Notre maquette n'est qu'un aspect de INGUIT, chacun étant libre de véhiculer par exemple la ligne MUTE par un quatrième fil dans la ligne «micro», ou toute autre exploitation.
Comme tel il ne nécessite aucun réglage, mais si d'aventure on souhaitait mettre en série plusieurs ensembles IC2, il faudrait tenir compte du diviseur R-ph total - R4, au besoin réduire R4 ou (et ?) réduire R5.
Pourtant il serait nettement plus élégant de grouper les commandes sur un seul IC2, et si nous n'avons prévu qu'un emplacement donc une seule ligne de commande, c'est un peu dans ce sens.
En effet, toutes les propositions faites précédemment tendent vers un affaiblissement de la modulation amplifiée par IC1A : mute, gate ou vibrato peuvent donc parfaitement obéir à une même commande de IC2, au prix d'un minimum de logique.

LESLIE ET PHASING

Voici un insert bien connu de tous, et pour lequel nous n'allons pas donner une solution mais plusieurs: de la commande manuelle au cycle automatique, en passant par la gestion I2C ou I/O PC, tout sera envisageable. Chacun optera alors pour la formule qui répondra au mieux à ses exigences.
Cet effet consiste à déphaser certaines fréquences pour les valoriser quand elles sont en phase avec le signal source (somme), ou les défavoriser quand les phases s'opposent (différence).
Un peu rapidement, on a voulu faire croire que le phasing pouvait supplanter l'effet LESLIE (notamment celui des fameuses 760 HAMMOND).

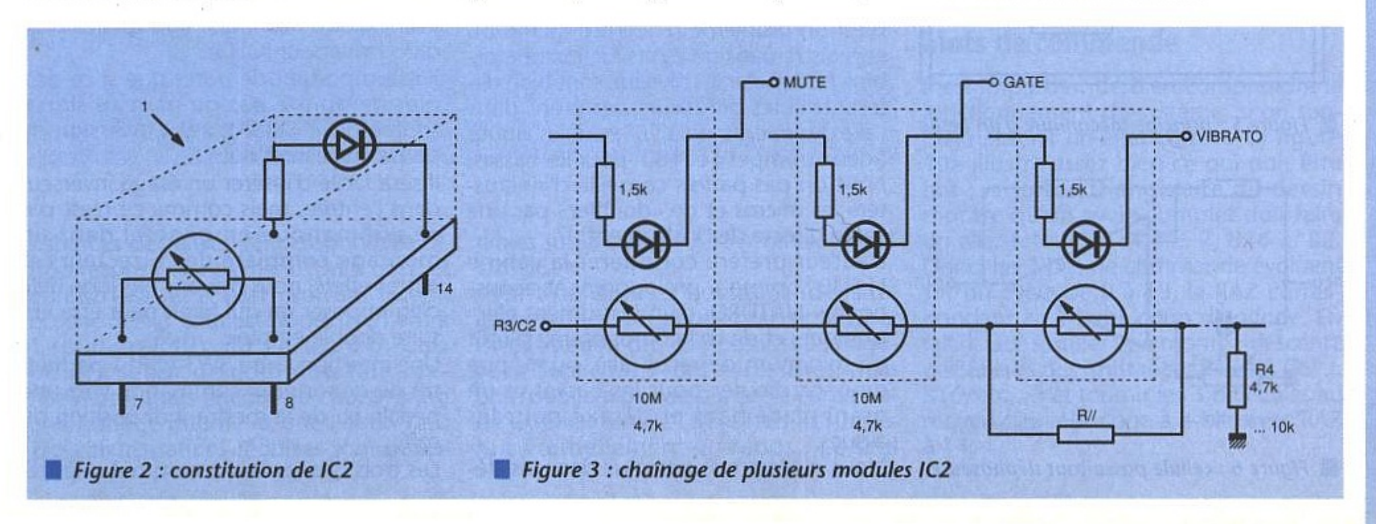
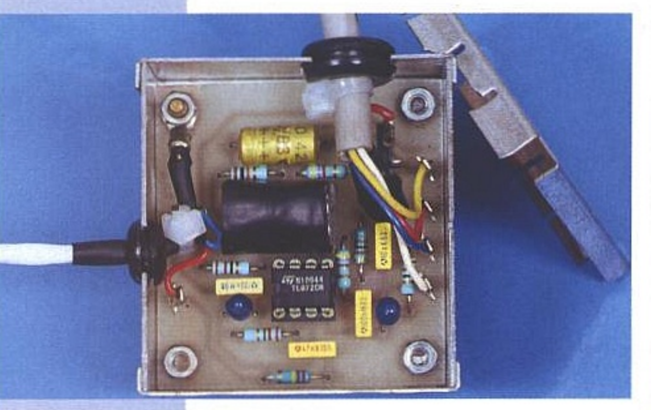
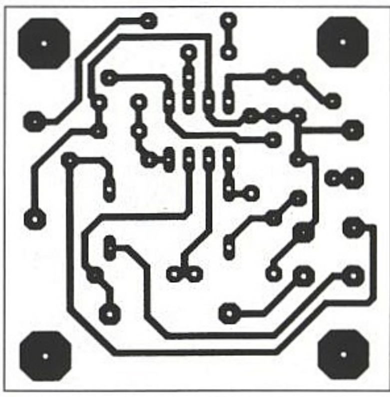


Figure 2 : constitution de IC2

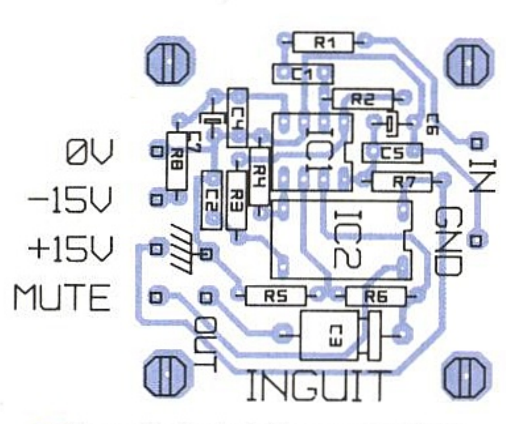
Figure 3 : chaînage de plusieurs modules IC2







■ *Figure 4a : CI Inguit*

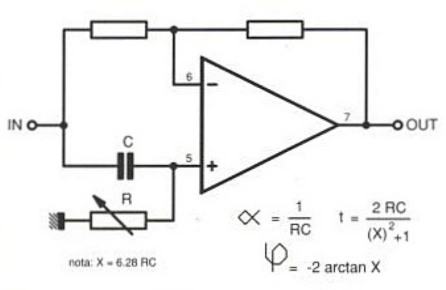


■ *Figure 4b : implantation correspondante*

La **figure 5** montre l'assemblage mécanique de cette machine.
Un diffuseur en polystyrène tourne au dessus du haut parleur de basses, alors que ce sont les trompes de la chambre de compression qui assurent la projection des aigues. Les deux moteurs disposent chacun de 2 vitesses, et on peut imaginer le résultat acoustique d'un tel ensemble, fort difficile à imiter avec un montage électronique quand on le connaît bien.
En effet, les phénomènes d'accélération et de décélération des éléments tournants interviennent de manière considérable sur le son obtenu. Sachant que les éléments sont de masses différentes, que les liaisons moteurs se font par courroies textile (avec tous les glissements qu'une telle solution suppose), il n'est pas difficile d'évaluer la complexité des phénomènes à reproduire, surtout si on sait que les changements de vitesses ne sont pas faits par commandes électriques des moteurs, mais par embrayage ou non d'un «frein».
Il y a eu à ce sujet diverses techniques,

double trompe
MOTEUR 1
AIGUES
MOTEUR A CHAMBRE DE COMPRESSION
MOTEUR 2
DIFFUSEUR
HP BASSES

■ *Figure 5 : principe mécanique d'un Leslie*



■ *Figure 6 : cellule passe-tout déphaseuse*

comme un galet électro-commandé sur deux axes moteurs tournant à des vitesses différentes, mais les résultats n'ont jamais atteints ceux de la solution HAMMOND.
C'est dire si la seule imitation de l'effet Doppler avec un réglage de vitesse aux changements instantanés, est bien loin de la vérité !
Aussi nous a-t-il semblé intéressant de proposer une carte qui pourrait être pilotée par soft, à charge alors de se casser la tête pour gérer les accélérations, décélérations, l'influence des tensions de courroies, etc.., voire commander deux cartes et travailler en deux voies actives !
Pour info, l'inertie de l'élément tournant des aigues est bien plus grande que celle des basses: les trompes mettent plus de temps pour passer de vitesse lente à rapide (entre 2 à 5 secondes) et le retour est de l'ordre de 2 secondes, suite à un violent coup de «patin».
Le diffuseur des basses étant très léger, les changements sont beaucoup plus rapides (notamment au freinage), et souvent les musiciens demandent de détendre la courroie du moteur concerné, afin de créer un temps mort par glissement sur les poulies. Ceci a pour effet de retarder un peu la prise de vitesse ainsi que le freinage, afin de tendre vers une pseudo-synchro des graves et des aigues.
Néanmoins, il ne faudrait pas assimiler temps mort puis prise de vitesse rapide (quand la courroie « colle enfin » aux poulies), et accélération progressive ...
Ah ce n'est pas simple, et tout n'est pas encore réglé : nous avons parlé de vitesses différentes, mais pas de l'arrêt !
Et là, tout est possible car rien n'est prévu pour que le diffuseur ou les trompes se reposent à des endroits précis, et connus...
C'est un problème rencontré quand on enregistre dans de petits locaux une cabine LESLIE, car les niveaux sont bien différents si les diffuseurs s'arrêtent dans l'axe des micros, ou à 90° pour les aigues (deux trompes) et 180° pour les basses.
N'a-t-on pas parfois conseillé d'«incruster» les micros et de «doubler» par une sortie directe de l'instrument..!?
L'auteur préfère considérer la cabine LESLIE comme « prolongement acoustique » NATUREL d'un instrument électronique, et de ce fait proposerait plutôt un minimum de recul (air), ou au pire deux cardioïdes pour les basses et un omni placé haut et désaxé pour les aigues.
Nous n'entrerons pas plus dans les détails, mais on peut constater qu'une simulation par soft mérite de tenir compte de nombreux «détails». Un PLUS consisterait encore à revenir en position FLAT (en douceur), c'est à dire après avoir stoppé la «rotation» au bon endroit, pour pouvoir la relancer également de façon discrète.
La circuiterie classique d'un phasing fait appel aux services des filtres «passe-tout», lesquels ont pour particularité de ne pas toucher à la bande passante, mais de modifier la phase à la fréquence pour laquelle ils ont été calculés.
La **figure 6** présente une cellule passe-tout à déphasage différé, couramment utilisée pour produire l'effet Doppler quand R varie et que plusieurs cellules se répartissent la bande utile.
Pour C donné on obtient un déphasage de 90° à f = 1/6,28 RC. En cascadant plusieurs cellules décalées d'une octave dont les résistances R évoluent en harmonie, une « rotation de phase » conduit - après mélange avec le signal source - à une impression de «rotation du son».

Schéma

Le schéma que nous proposons est visible **figure 7**. Il se constitue d'un étage tampon d'entrée (IC10B) , suivi de six cellules identiques à la figure 6 (décalées chacune d'une octave) et dont R est sélectionnée parmi 8 valeurs par un mot de 3 bits commandant des commutateurs 4051 (IC4 à IC9). Les 8 combinaisons «figées» permettent d'obtenir un effet très satisfaisant directement gérable par soft, contrairement à la méthode faisant appel à des transistors à effet de champ préalablement apariés, ou à des AM9709 devenus rares.
La sortie des filtres est dosable par P1 avant d'être mélangée au signal source dans l'inverseur IC10A.
Il est important de noter que si ce dernier ne donne pas de gain au signal source (R52 = R54), il y a inversion de phase entre in et out.
Il sera facile d'insérer un étage inverseur dans l'entrée, mais comme ce n'est pas ce qui manque en général dans un montage complété de correcteurs et autres effets, nous n'avons pas jugé utile d'en imposer un qui ferait peut être ensuite double emploi.
Une insertion entre SW1 et RL1 permettra de déporter la commande vers une pédale ou de la mettre à disposition du soft.
Les trois bits servant à commuter les ré-



■ *Figure 7 : schéma du Phasing*

sistances des filtres sont disponibles de trois façons : directement (Data out), et en entrée A et A barre suivant que la broche Select data est respectivement à 1 (pull up R49), ou portée à 0. Ceci pourra servir de commutation manuel/auto, mais encore de choisir entre une horloge interne et une liaison I2C. Trois alimentations régulées sont disponibles sur la carte : +/-15 V (analog) et 5 V digit.

Avant de donner quelques idées de commandes, on peut observer le CI et l'implantation des composants à la **figure 8**. Sept straps sont nécessaires, mais on n'oubliera pas de ponter la commande extérieure (vers Ld1) sinon RL1 resterait éternellement au repos. Le seul point délicat de cette construction consiste à bien placer les 48 premières résistances. Chaque cellule étant identique (aux condensateurs près), nous avons fait un zoom (**figure 9**) pour qu'il n'y ait pas de confusion possible.

Ici, il ne s'agit que de IC 4 (première cellule), donc des résistances R1 à R8. Si l'ordre n'est pas logique, c'est uniquement pour respecter les broches des 4051 et simplifier le dessin du CI.

La meilleure méthode consistera à monter les six résistances de mêmes valeurs aux mêmes emplacements, autour d'IC 4 à IC 9 (voir nomenclature).

Mots de commande

Il est important de bien comprendre le fonctionnement du système, si on souhaite obtenir un effet correct. La **figure 10a** illustre assez bien ce qui doit être fait : la forme triangulaire du dessin montre qu'un cycle complet doit faire un aller-retour tel : R1 ... 7, 8, 6 ... R2. Donc, les 3 bits de commande évoluent sur un cycle de 0 à 13, la RAZ correspondant au 14ème coup d'horloge. En fait, il faut simuler une montée-descente des valeurs des résistances R1 à R8 (R9 ... R16, etc ...) et fournir les 3 bits utiles au moyen d'une horloge à 4 bits avec RAZ à 14.



La **figure 10b** est une proposition Hard avec le dump de l'Eprom correspondant à une exploitation simple.
Elle est un peu «lourde», mais facile à comprendre = une horloge vient incrémenter un 7493 dont les 4 bits de sortie adressent une Eprom, laquelle impose à D0/D2 les 14 données irréductibles, puisque la RAZ est détectée sur les adresses et force le 7493.
Une suggestion d'affichage donnant une image de l'effet a été ajoutée pour le «fun» : un 74154 récupère les adresses et pilote une LED parmi 14, l'idéal étant alors de les implanter en cercle afin de constituer une roue qui indique si on incrémente ou décrémente.
Une telle construction fera peut être sourire certains d'entre vous, mais ce n'est qu'une suggestion pouvant convenir à une commande automatique ou manuelle ordinaire, sans oublier pourtant qu'elle libère considérablement une gestion soft, en ne lui laissant à traiter que les tops d'horloge (avec récupération de la RAZ).
Si on se rappelle ce qui a été dit des caractéristiques particulières à un effet LES-

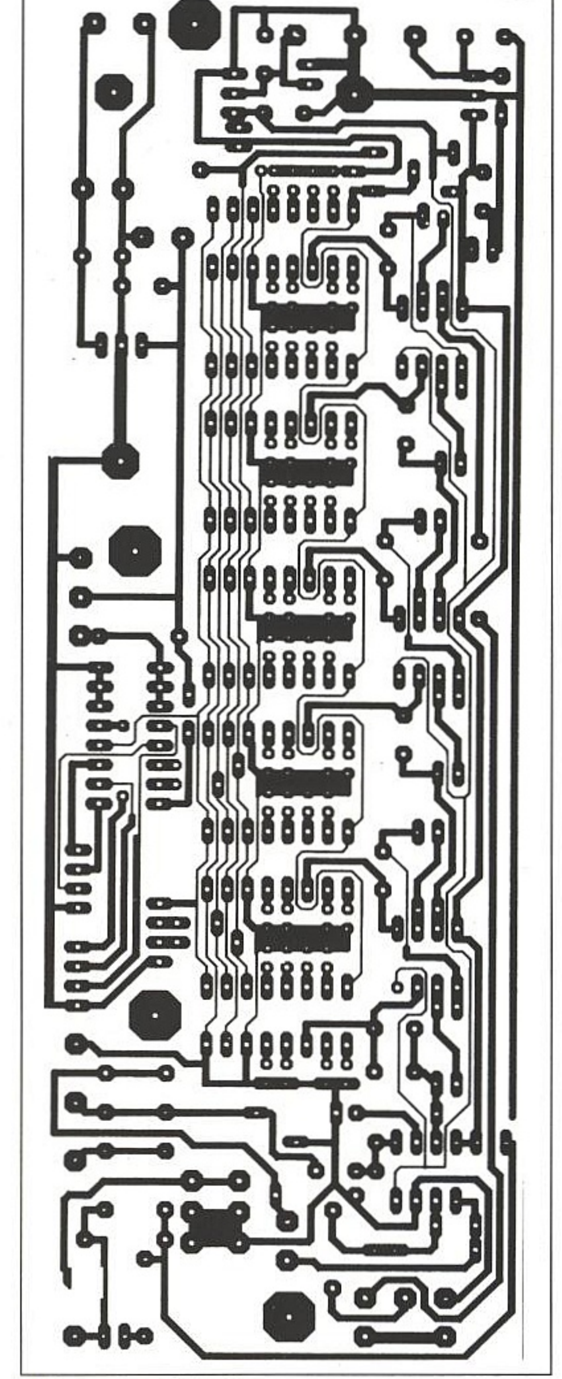


Figure 8a : CI Phasing

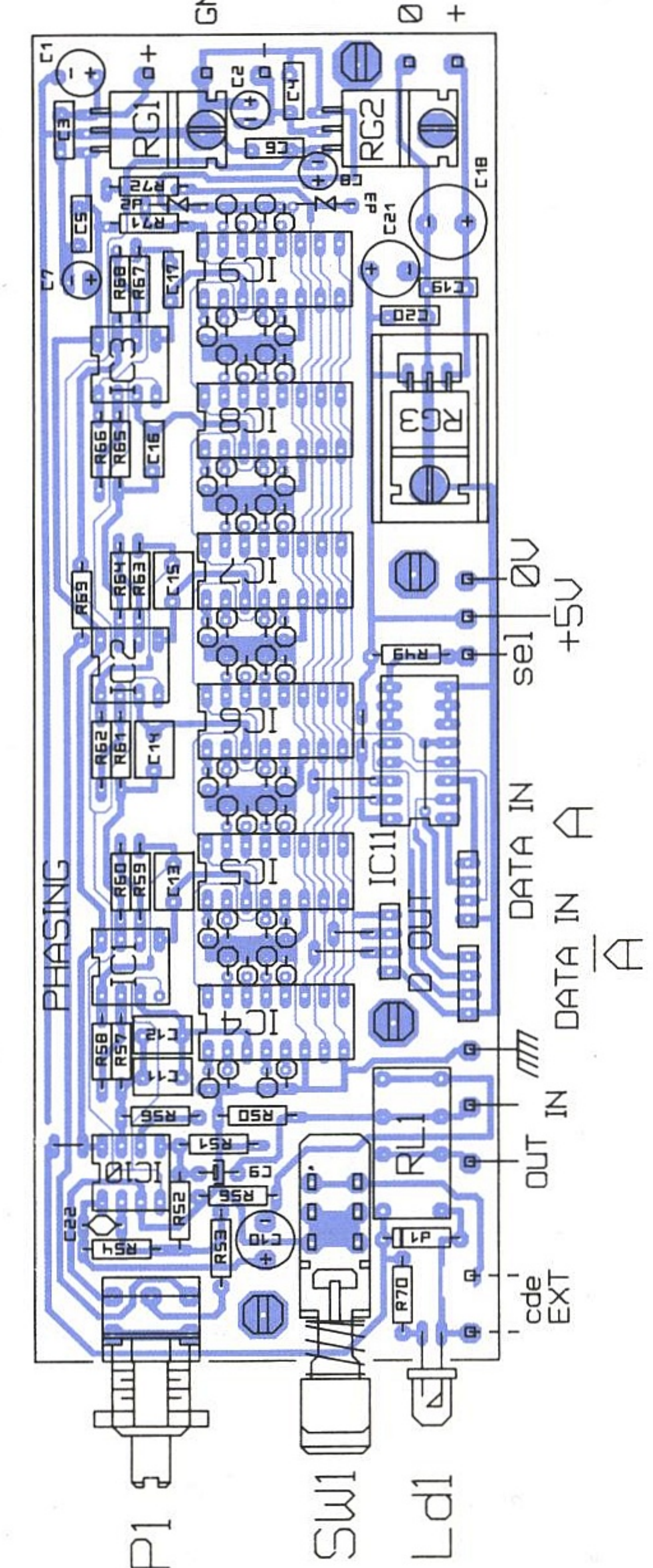


Figure 8b : implantation Phasing

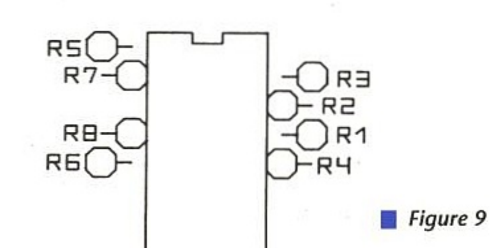


Figure 9

Nomenclature

INGUIT

Résistances

R1 : 220 kΩ
R2 : 2,2 MΩ
R3 : 1 kΩ
R4, R5 : 4,7 kΩ
R6 : 10 kΩ
R7, R8 : 22 Ω

Condensateurs :

C1 : 47 nF MILFEUIL
C2 : 10 nF MILFEUIL
C3, C6, C7 : 10 µF 25 V
C4, C5 : 100 nF MILFEUIL
C11 à C13 : 470 nF MILFEUIL

Semi-conducteurs :

IC1 : TL072 + Support
IC2 : LDR + LED 5 mm rouge + résistance 1,5 kΩ + support 14 broches

Divers :

Boîtier TEKO ref. 371
Cosses, passes-fils, visserie

PHASING

Résistances:

R1, R9, R17, R25, R33, R41 : 100 Ω
R2, R10, R18, R26, R34, R42 : 220 Ω
R3, R11, R19, R27, R35, R43 : 330 Ω
R4, R12, R20, R28, R36, R44 : 470 Ω
R5, R13, R21, R29, R37, R45 : 1 kΩ
R6, R14, R22, R30, R38, R46 : 2,2 kΩ
R7, R15, R23, R31, R39, R47 : 3,3 kΩ
R8, R16, R24, R32, R40,= R48 : 4,7 kΩ
R49 : 10 kΩ
R50, R51, R55 : 47 kΩ
R52, R54 : 100 kΩ
R53, R56 à R68 : 22 kΩ
R69 : 10 MΩ
R70 à R72 : 1 kΩ

Condensateurs:

C1, C10, C21 : 100 µF 25V
C2 : 22 µF 25V
C3 à C6, C16, C19, C20 : 100 nF MILFEUIL
C7 à C9 : 10 µF 25V
C11 à C13 : 470 nF MILFEUIL
C14 : 330 nF MILFEUIL
C15 : 150 nF MILFEUIL
C17 : 47 nF MILFEUIL
C18 : 220 µF 25V

Semiconducteurs:

RG1 : 7815
RG2 : 7915
RG3 : 7805 + radiateur
D1 : 1N4007
D2, D3 : Zener 7,5V
Ld1 : LED rouge 5 mm
IC1 à IC3 : TL072
IC4 à IC9 : CD4051
IC10 : NE5532
IC11 : CD4053

Divers:

P1 : 100 kA P11
SW1 : Schadow 2 inv
RL1 : Relais 12V MR62 ou eq.
Supports IC : 4 de 8 broches, 7 de 16 broches
3 connecteurs HE14 4 points , cosses et visserie.

LIE réel, il y a encore de quoi jouer rien qu'à vouloir simuler correctement les sept cas de figures classiques :
1/ Stop (sans déphasage)
2/ de Stop à vitesse lente
3/ de Stop à vitesse rapide
4/ de vitesse lente à Stop
5/ de vitesse rapide à Stop
6/ de vitesse lente à rapide
7/ de vitesse rapide à lente
Bien entendu, un doublage des conditions pour un système à 2 voies serait idéal, et consisterait à travailler déjà sur 2 sorties d'horloges, 2 entrées de RAZ, et 3 entrées d'ordre utilisateur Stop, Slow, Fast.

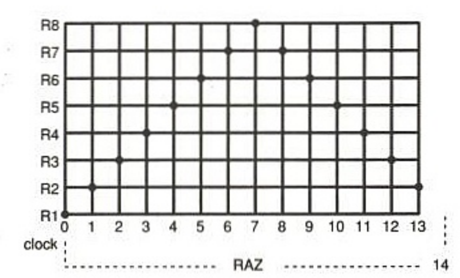


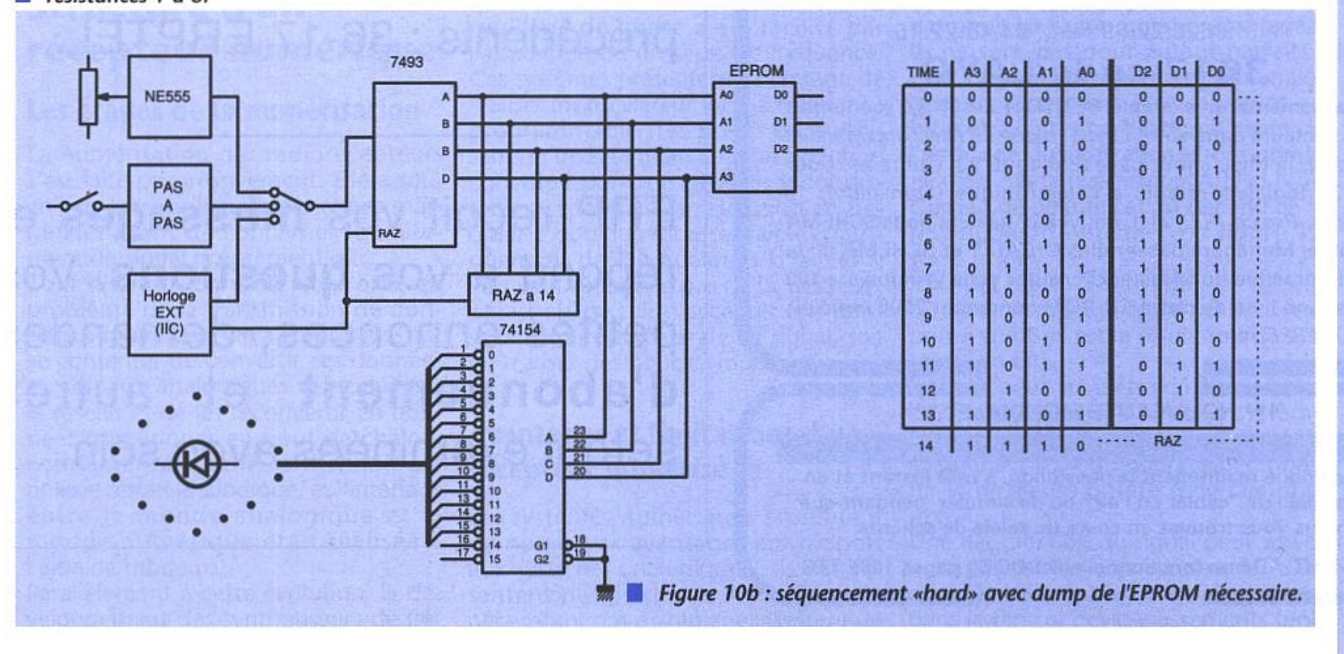
Figure 10a : balayage temporel des résistances 1 à 8.

En I2C, un PCF 8574 conviendrait parfaitement quitte (une fois les algorithmes bien au point) à dérouter les data d'horloges et de RAZ pour ne plus gérer que six données en sorties (D0 - D2 Low, High) et deux entrées commandées respectivement par Start/Stop et Slow/Fast. Chacun fera comme bon lui semble, mais une pédale de commandes (ou un report sur des touches clavier) des trois informations distinctes Stop, Slow et Fast, est nettement plus confortable.
Si on souhaite également reporter un affichage en cercle, un second PCF 8574 s'impose alors. Ne serait-il donc pas judicieux de partager harmonieusement Hard et Soft, afin de laisser au hard les tâches laborieuses et répétitives, au profit d'un soft plus élaboré (et plus rapide) réservé à la simulation mécanique, ici très importante ?

CONCLUSION

Si INGUIT est immédiatement utilisable (même sur casque 600 Ohms avec un très bon niveau d'écoute), la carte PHASING attend des mots de commandes pour fonctionner. Passés les premiers essais en manuel avec une roue codeuse, il faudra ensuite s'attaquer à un pilotage plus ou moins performant, mais dans tous les cas passionnant. Pour l'instant l'auteur travaille avec COMMnet et un PCF 8574 (plus l'aide de l'ESPION, ERP n°552), et quand le soft sera parfaitement en mesure de reproduire les comportements mécaniques d'une 760, il sera temps d'affiner encore en doublant les 4051 pour « lisser les pentes », et obtenir alors une VRAIE LESLIE portative.
Bon courage !

Jean ALARY



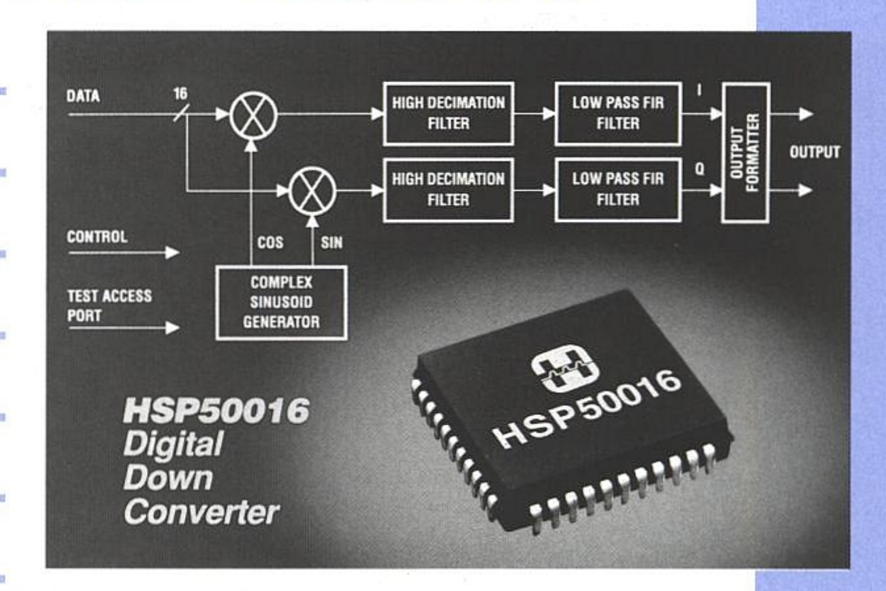
TIME	A3	A2	A1	A0	D2	D1	D0
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	1
4	0	1	0	0	1	0	0
5	0	1	0	1	1	0	1
6	0	1	1	0	1	1	0
7	0	1	1	1	1	1	1
8	1	0	0	0	1	1	0
9	1	0	0	1	1	0	1
10	1	0	1	0	1	0	0
11	1	0	1	1	0	1	1
12	1	1	0	0	0	1	0
13	1	1	0	1	0	0	1
14	1	1	1	0	RAZ		

Figure 10b : séquencement «hard» avec dump de l'EPROM nécessaire.



La réception radio numérique Étude du HSP 50016

C'est bien connu, les techniques numériques sont de plus en plus utilisées en électronique, que ce soit dans le domaine de la basse fréquence ou de la vidéo. Pourtant, la radio est longtemps restée à l'écart de ce mouvement, sans doute parce que les fréquences mises en jeu se situent au-delà du domaine de couverture des DSP classiques qui dépasse difficilement le MégaHertz. Progressivement, de nouveaux processeurs numériques, conçus à partir d'une logique cablée plutôt que programmable, sont apparus. Parmi ces circuits suffisamment rapides pour les applications radio, le HSP50016 de HARRIS Semiconductor peut constituer le coeur d'un récepteur de radiocommunications.



Structure d'un récepteur numérique

Les étapes de la numérisation

La numérisation des radiorécepteurs s'est faite progressivement. Elle a touché en premier lieu des fonctions marginales avant de concerner le traitement de signal proprement dit.
Tout d'abord, on a été confronté au problème de la transmission de données numériques. On a commencé par se contenter de convertir ces données en signaux analogiques véhiculés par le réseau et de les reconvertir en données numériques en bout de chaîne. Fondamentalement, la technologie utilisée restait analogique, et l'interface entre le monde analogique et le monde numérique était réalisée à l'aide de modems.
Parallèlement à cette évolution, le développement des synthétiseurs de fréquence utilisant des boucles à verrouillage de phase a été facilité par l'apparition de diviseurs de fréquence. Ces systèmes présentent l'avantage de réaliser un oscillateur local à fréquence programmable très précis en n'utilisant qu'un seul quartz. Cette évolution constitue donc un progrès incontestable. Cependant la résolution en fréquence obtenue est assez faible (classiquement de 1 à quelques kHz) car si l'on veut réduire le pas de la synthèse, on est obligé soit d'accepter une baisse de la pureté spectrale du signal, soit de réaliser des boucles multiples.

Avantages et limitations d'un récepteur numérisé

Les systèmes numériques présentent de nombreux avantages par rapport aux systèmes analogiques : ils ne présentent pas de dérive thermique, ne nécessitent pas de réglage, assurent un filtrage efficace, peuvent être reprogrammés de façon quasi- instantanée. Ils ne sont pas pour autant parfaits : tout comme leurs homologues analogiques, il peuvent être sensibles au bruit et à l'intermodulation à cause de l'erreur de quantification et des non-linéarités des convertisseurs A/N et N/A.
L'erreur de quantification dépend de la résolution du système (voir **figure 1**), selon la loi :

$$R = 6{,}02\,n + 1{,}76 \quad \text{(dB)}$$

où : R est le rapport signal sur bruit, exprimé en dB, n est la résolution du système, en bits.
Le bruit peut aussi être dû à une instabilité de phase de l'horloge système (spectre étalé). Il est donc essentiel de soigner le circuit d'horloge afin d'obtenir la meilleure pureté spectrale possible.
Une dégradation du signal peut survenir au niveau du convertisseur N/A. En effet, lorsque le signal converti varie dans le temps, des dépassements fugi-



Figure 1 : bruit de quantification.

Figure 2 : modulation QAM.

Figure 3 : représentation complexe d'un signal.

Figure 4 : récepteur numérisé à conversion directe.

tifs (glitches) peuvent apparaître lors des transitions de niveau. Il en résulte l'apparition d'harmoniques au niveau du signal analogique reconstitué. L'importance de ce défaut dépend de la conception du convertisseur. Afin de pouvoir apprécier le phénomène, on caractérise le convertisseur par la marge dynamique entre signal utile et signaux indésirables (paramètre SFDR : Spurious Free Dynamic Range). Plus cette grandeur, exprimée en dB, est élevée, meilleur est le convertisseur.

D'autre part, les systèmes numériques sont assez gourmands en énergie, ce qui constitue un sérieux inconvénient dans le cas d'applications portables alimentées par batterie.

Malgré tout, les défauts affectant la qualité du signal sont facilement maîtrisables et les systèmes numériques sont sensiblement plus performants que leurs homologues analogiques. Le filtrage est plus efficace, le rapport signal/bruit est plus élevé et le risque d'intermodulation est considérablement réduit.

Modulations numériques

La modulation consiste à faire varier l'un des paramètres d'une porteuse afin de transmettre une information. On peut agir soit sur son amplitude, soit sur sa phase, soit sur les deux. Une modulation est d'autant plus efficace qu'elle permet de transmettre une grande quantité d'informations avec un étalement spectral réduit. Le débit d'une modulation numérique est exprimé en bits/s et et en Bauds. Ces deux grandeurs ne sont pas toujours égales. En effet on exprime en Bauds le nombre de symboles transmis par seconde. Or, le nombre de bits représentés par un symbole dépend du type de modulation.

Dans le domaine de la transmission de données numériques, les principaux types de modulation sont les suivants :

ASK : modulation en amplitude, par absence ou présence de la porteuse.

PSK : modulation en phase, par déphasage de 0 ou 180 degrés (inversion).

FSK : modulation en fréquence de la porteuse.

QPSK : variante de la PSK, à 4 états

FFSK : variante de la FSK, utilisée dans les modems radio 1200 Bauds.

GMSK : variante de la FSK et de la PSK, optimisée en occupation spectrale, utilisée dans le réseau GSM.

QAM : modulation très puissante, agissant à la fois sur l'amplitude et la phase. On peut avoir jusqu'à 64 états par caractère (voir **figure 2**)

Certes, un récepteur numérisé peut traiter indifféremment les signaux en modulation analogique et les signaux en modulation numérique. Cependant, la modulation numérique est préférable car elle présente une très grande souplesse quant à la nature des données transmises. Elle intègre presque toujours des mécanismes de correction d'erreur, ce qui constitue un avantage décisif.

Représentation réelle et représentation complexe

Les nombres complexes sont un excellent outil pour la représentation des signaux modulés. La porteuse peut être représentée par un vecteur d'amplitude constante, en rotation constante à la vitesse angulaire $2\pi f$ (rd/s). Le signal modulé peut être représenté par un vecteur d'amplitude A, tournant à la vitesse angulaire $2\pi(f + F)$ (rd/s), avec un angle initial égal à la phase φ.

On peut imaginer les projections de ces vecteurs sur les axes réel et imaginaire du plan complexe (voir **figure 3**).

Considérons une porteuse :

$p(t) = \cos 2\pi ft$ modulée par un signal :

$m(t) = A \cos (2\pi Ft + \varphi)$

Ces signaux ont une représentation complexe :

$P(t) = \exp (j2\pi ft)$
$= \cos 2\pi ft + j \sin 2\pi ft$

$M(t) = A \exp (j\ 2\pi Ft + \varphi)$
$= A [\cos(2\pi ft + \varphi)$
$+ \sin(2\pi Ft + \varphi)]$

Après passage dans le modulateur, on obtient :

- en représentation réelle :

$p(t) \bullet m(t) = A \cos 2\pi ft \bullet \cos (2\pi Ft + \varphi)$
$= (A / 2)[\cos (2\pi(f + F)t)$
$+ \cos (2\pi(f - F)t)]$

- en représentation complexe :

$P(t)\ M(t) = A \exp (j2\pi ft) \exp (2\pi Ft + \varphi)$
$= A \exp j\ [2\pi(f + F)t + \varphi]$

On constate qu'un signal modulé comporte 2 bandes latérales en représentation réelle, contre une seule bande latérale en représentation complexe. Cette caractéristique peut être utilisée avec profit dans les récepteurs numériques.

Récepteur numérisé à conversion directe

Il s'agit de la transposition numérique du récepteur à conversion directe classique. Sa structure apparaît en **figu-**



re 4. L'étage d'entrée reste analogique ; le facteur de bruit du système dépend essentiellement de ses performances. Il doit être associé à un filtre afin d'éliminer les signaux indésirables. Il peut être suivi d'un autre étage analogique à gain variable. la commande automatique de gain est destinée à présenter à l'entrée du convertisseur un signal permettant d'utiliser convenablement la dynamique permise par la résolution choisie.
A ce niveau, il semble qu'une résolution de 12 bits soit un minimum pour les applications classiques. Actuellement, les performances des meilleurs convertisseurs monolithiques disponibles sur le marché se situent aux alentours de 10 MSPS (millions d'échantillons par seconde) pour une résolution de 14 bits et de 3 MSPS pour 16 bits. En tenant compte du critère de Nyquist ou de Shannon (afin d'éviter le recouvrement des spectres, la fréquence d'échantillonnage doit être au moins le double de la bande passante du signal échantillonné), il est possible de recevoir par cette méthode des signaux de quelques MégaHertz.
Le signal numérisé est ensuite mélangé dans un multiplieur avec un signal issu d'un oscillateur à commande numérique (NCO : Numerically Controlled Oscillator). Le signal issu du battement est ensuite énergiquement filtré pour restituer la modulation d'origine.

Récepteur superhétérodyne à double conversion de fréquence

Le récepteur à conversion directe, même numérique, n'est utilisable que pour la réception de fréquences relativement basses. Il est excellent pour recevoir les signaux horaires que l'on trouve en-dessous de 150 kHz, car le filtrage numérique permet la réception à bande très étroite. Cependant, lorsque la fréquence à recevoir s'élève, une autre configuration devient nécessaire pour garder de bonnes performances.
La **figure 5** nous présente la structure d'un récepteur à double changement de fréquence, adapté aux VHF. Le signal est d'abord traité par des étages analogiques : il est filtré et amplifié par un amplificateur sélectif, puis transposé sur une première fréquence intermédiaire (10,7 MHz) pour assurer une bonne réjection de la fréquence image. Après amplification et filtrage, le signal est ensuite retransposé sur une deuxième fréquence intermédiaire (455 kHz) afin de pouvoir utiliser un convertisseur A/N 16 bits. On pourrait se contenter d'une seule fréquence intermédiaire, car les filtres numériques sont suffisamment efficaces pour assurer une bonne sélectivité à 10,7 MHz, mais on serait obligé d'utiliser des convertisseurs à faible résolution et les performances du récepteur seraient moins bonnes.
Le signal à 455 kHz est numérisé et attaque 2 mélangeurs qui reçoivent d'autre part 2 signaux en quadrature issus d'un NCO.
En sortie des mélangeurs, après filtrage sur chacune des 2 voies, on trouve une représentation complexe du signal démodulé. Cette configuration, compliquée en apparence, permet de démoduler aussi bien les signaux modulés en phase qu'en amplitude. Elle sera détaillée dans la description du HSP50016.

Récepteur UHF numérisé

La structure globale d'un tel récepteur est présenté à la **figure 6**. Dans le cas du réseau de radiotéléphone numérique paneuropéen GSM, il faut pouvoir recevoir une bande de fréquence large de 35 MHz, située aux alentours de 900 MHz. On transpose donc le signal à recevoir sur une fréquence intermédiaire d'au moins 70 MHz (75 MHz par exemple) pour éviter d'être perturbé par la fréquence image.
La bande passante de l'étage d'entrée (qui fonctionne en large bande) est fixée à 35 MHz par un filtre à ondes de surface (SAW filter). La préamplification et la conversion en fréquence du signal d'entrée peuvent être réalisées

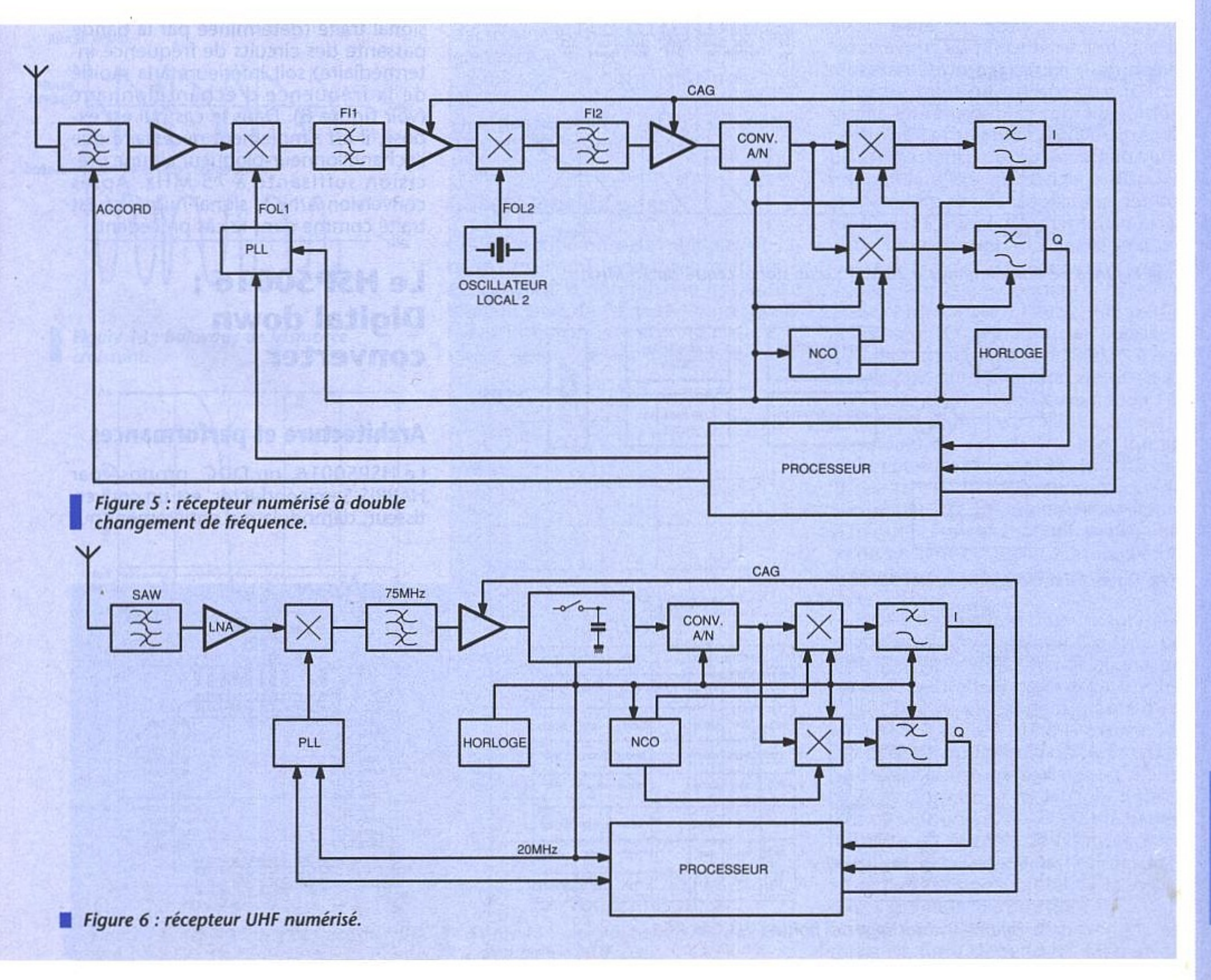


Figure 5 : récepteur numérisé à double changement de fréquence.

Figure 6 : récepteur UHF numérisé.

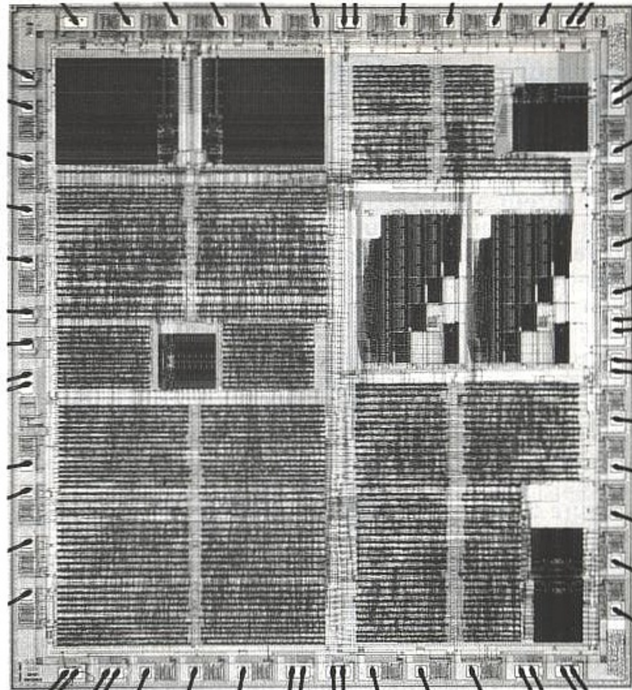


Figure 11 : aspect de la puce du HSP50016.

par un circuit intégré unique, tel que le HFA3600 de HARRIS Semiconductor (voir **figure 7**). A 900 MHz, la partie préamplificateur de ce circuit intégré présente un facteur de bruit de 2,4 dB et un point d'interception de troisième ordre correspondant à +10,7 dBm en sortie, ceci pour un gain de 11 dB. En ce qui concerne le mélangeur, ces valeurs sont respectivement de 6,7 dB, +13,1 dBm et 8,1 dB. Lorsque les deux étages sont combinés, on obtient un facteur de bruit de 3,1 dB et un IP3 situé à +12,1 dBm en sortie, pour un gain global de 19,1 dB.

Une fréquence intermédiaire de 75 MHz est encore élevée pour les circuits numériques actuels, à moins d'utiliser des technologies avides d'énergie, comme l'ECL. La solution à ce problème réside dans l'échantillonneur-bloqueur qui précède le convertisseur A/N. En effet, cet étage peut être assimilé à un multiplieur dont l'une des entrées est attaquée par un signal en tout ou rien, riche en harmoniques. Avec une fréquence d'échantillonnage de 20 MHz, il y a battement entre l'harmonique 2 (40 MHz) et le signal reçu, ce qui donne naissance à un signal dont le spectre (autour de 40 - (75 - 40) = 5 MHz) est le symétrique du signal d'entrée. Le signal résultant pourra être traité par la chaîne numérique.

En dépit des apparences, cette solution respecte le critère de Nyquist. Il suffit pour cela que la largeur du spectre du signal traité (déterminée par la bande passante des circuits de fréquence intermédiaire) soit inférieure à la moitié de la fréquence d'échantillonnage (voir **figure 8**). Dans le cas qui est exposé, il est simplement nécessaire que l'échantillonneur-bloqueur ait une précision suffisante à 75 MHz. Après conversion A/N, le signal numérisé est traité comme dans le cas précédent.

Figure 7 : le HFA3600.

Figure 8 : échantillonnage à 20 MSPS d'un signal centré sur 75 MHz.

Le HSP50016 : Digital down converter

Architecture et performances

Le HSP50016, ou DDC, proposé par HARRIS Semiconductor, est un convertisseur, démodulateur entièrement nu-

Figure 9 : architecture du HSP50016.

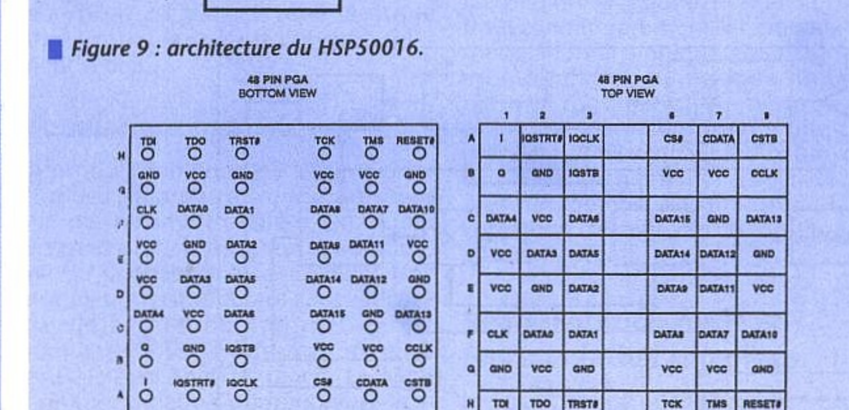


Figure 10 : brochage des boîtiers PLCC et PGA.



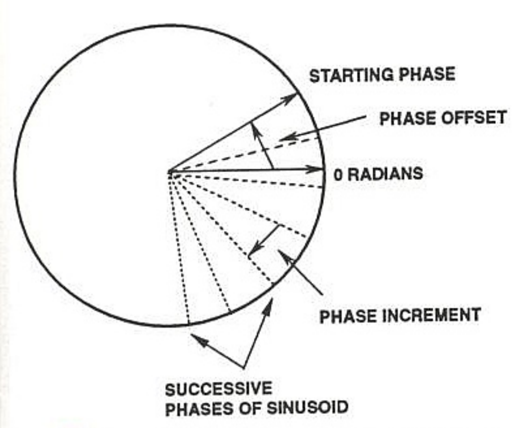
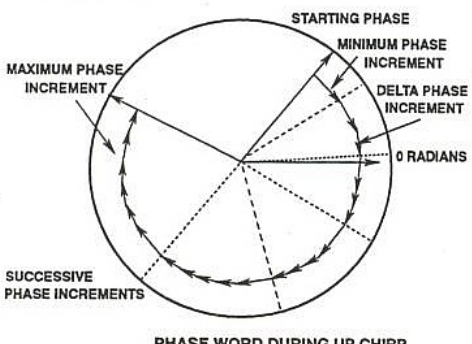


Figure 12 : fonctionnement du générateur de phase.



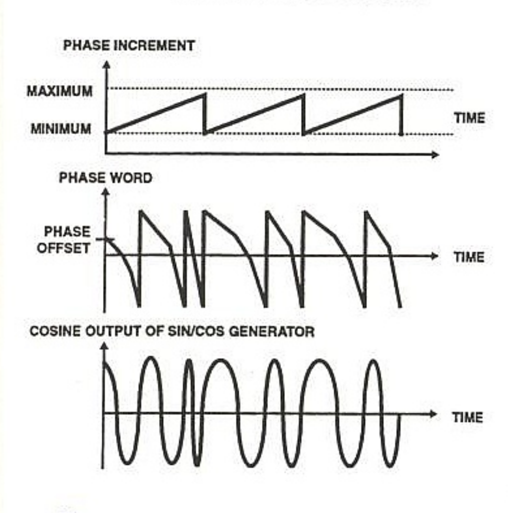


Figure 13 : balayage de fréquence croissant.

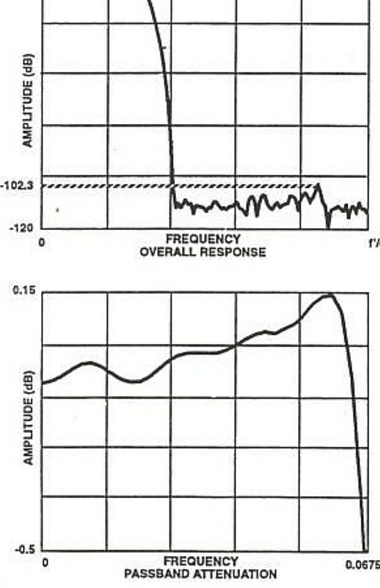


Figure 14 : réponse en fréquence du HDF seul.

mérique réalisé en technologie CMOS utilisable jusqu'à une fréquence d'échantillonnage de 52 MHz. Il est disponible au choix dans un boîtier PGA de 48 broches ou dans un boîtier PLCC de 44 broches.

Comme le montre la **figure 9**, il est constitué d'un générateur de phase très sophistiqué capable de fonctionner suivant plusieurs modes : fréquence fixe, saut de fréquence, balayage croissant, balayage décroissant. Ce générateur commande un générateur sinusoïdal complexe dont les 2 signaux en quadrature sont mélangés avec le signal d'entrée dans un modulateur. Il en résulte 2 signaux qui correspondent à la partie réelle et à la partie imaginaire de la représentation complexe du signal issu du mélange. Chacun est véhiculé sur une chaîne de traitement spécifique et subit un filtrage énergique. La plupart des paramètres définissant le fonctionnement du circuit sont programmés par écriture dans des registres de contrôle. Enfin, pour compléter l'ensemble, un port spécial permet le test sur carte du circuit intégré.

La technologie numérique de ce circuit intégré permet l'obtention de performances exceptionnelles :

- résolution en fréquence de l'oscillateur local : mieux que 0,006 Hz
- ondulation dans la bande passante : inférieure à 0,04 dB
- atténuation hors bande : supérieure à 106 dB
- facteur de forme (de -3dB à -102 dB) : inférieur à 1,5
- résolution du signal en entrée : 16 bits
- marge dynamique (SFDR) du modulateur : supérieure à 102 dB

Générateur de phase

Cette fonction offre de multiples possibilités : suivant la programmation, il est possible d'obtenir une fréquence fixe, un saut de fréquence, un glissement linéaire de fréquence dans le sens croissant, ou dans le sens décroissant. A une fréquence donnée, l'incrément de phase est égal à :

$d\varphi = 2^{33} \bullet Fc/Fs$

avec Fol fréquence de l'oscillateur local et Fs fréquence d'échantillonnage.

Si l'incrément de phase est constant, la fréquence Fol est constante. Cependant, le HSP50016 offre la possibilité de faire évoluer l'incrément de phase. A chaque cycle d'horloge, l'incrément de phase ("phase increment") peut être lui même incrémenté d'une valeur appelée "delta phase incrément". On peut ainsi le faire évoluer entre deux valeurs limites.

Concrètement, ceci se traduit par un balayage linéaire en fréquence (scanning). La **figure 13** montre l'évolution des paramètres pour un balayage de fréquence dans le sens croissant.

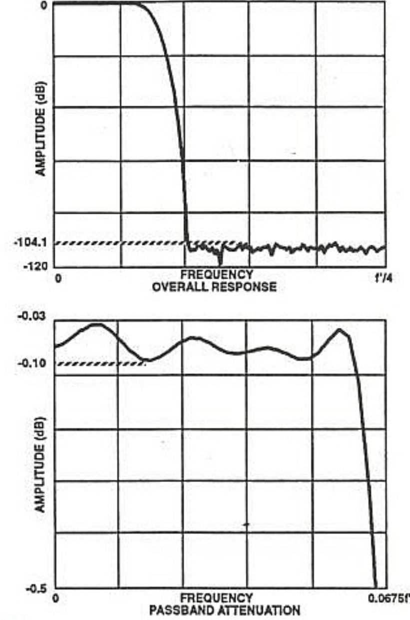


Figure 15 : réponse en fréquence de la combinaison HDF + FIR.

Générateur sinusoïdal et modulateur

Le générateur est une simple table de correspondance. A partir de la valeur de la phase, il produit deux signaux sinusoïdaux en quadrature, représentés sur 17 bits. En mélangeant séparément ces signaux au signal d'entrée dans le modulateur, on obtient une représentation complexe signal issu du battement. Le signal d'entrée est défini sur 16 bits, en binaire pur ou en complément à 2, selon la programmation du circuit.

Filtres passe -bas

Nous avons vu précédemment que la représentation complexe d'un signal modulé occupe un spectre 2 fois moins étendu que sa représentation réelle. Ceci permet de doubler la bande passante du système pour une fréquence d'échantillonnage donnée et d'étendre par conséquent le domaine d'application d'un circuit numérique limité en rapidité. Le signal est traité à travers deux canaux identiques, l'un correspondant à la partie réelle (I : In-phase), l'autre à la partie imaginaire (Q : Quadrature). Chacun de ces canaux est constitué de 2 étages : un filtre à rapport de décimation élevé (HDF : High Decimation Filter), suivi d'un filtre "FIR" à réponse impulsionnelle finie (FIR : Finite Impulse Response). Avant de décrire ces filtres dans le détail, précisons que la décimation est un processus par lequel la fréquence d'échantillonnage en sortie de système est égale à un sous-multiple de la fréquence d'échantillonnage en entrée.

Le HDF est constitué d'un intégrateur suivi d'un filtre en peigne. Ce type de filtre permet l'extraction d'un signal de bande très étroite à partir d'un spectre à large bande. La bande passante d'un tel filtre est représenté à la **figure 14**. Elle dépend du rapport de décimation du HDF qui est compris entre 16 et 32768. Comme le montre la figure, plus ce rapport est élevé, plus la bande passante est étroite. Ce filtre présente aussi une excellente réjection des harmoniques du signal présent à la sortie, ce qui diminue la distorsion.

L'atténuation du signal disponible en sortie de filtre dépend du facteur de

décimation. Elle est compensée à l'aide d'un multiplieur à facteur d'échelle programmable.
Le filtre FIR qui suit le HDF est destiné à diminuer les ondulations et à améliorer la réponse en fréquence dans la bande de transition. Il est le siège d'une décimation d'ordre 4 et son efficacité correspond à un filtre à 121 coefficients. La résolution en sortie de filtre est de 18 bits. La **figure 15** donne un aperçu de la réponse globale et permet de mesurer l'amélioration obtenue en faisant suivre le HDF par un filtre FIR.

Conversion sans inversion de spectre

On dispose à la sortie des filtres de 2 signaux en quadrature I et Q, qui correspondent respectivement à la partie réelle et à la partie imaginaire du signal reçu. La fréquence d'échantillonnage f" est définie par :
f" = f' / 4 = fs / (4 • R) où f" est la fréquence d'échantillonnage en sortie de FIR, f' la fréquence d'échantillonnage en sortie du HDF, et R le rapport de décimation du HDF.
Le spectre du signal complexe ainsi obtenu, centré sur la fréquence 0, s'étend entre -f"/4 et +f"/4 (voir **figure 16**). Ce signal a été obtenu en mélangeant avant filtrage le signal reçu avec un signal issu de l'oscillateur local de fréquence Fol. Cette fréquence Fol est centrée sur l'une des bandes latérales du signal reçu. Il est décrit par l'expression :

$$z(n) = \cos[2\pi(F - Fol)n] + j\sin[2\pi(F - Fol)n] = \exp[j2\pi n(F - Fol)]$$

Il est possible de convertir ce signal complexe en un signal réel, particulièrement lorsqu'on n'a pas besoin de l'information de phase (démodulation de l'AM, de la BLU, ou de l'ASK). Pour cela, le spectre du signal doit d'abord être décalé de f"/4. Cela s'obtient en multipliant le signal par un signal de fréquence égale au quart de la fréquence d'échantillonnage. Un signal correspond à la suite d'échantillons suivante :
1 + 0j, 0 + j, -1 + 0j, 0 - j, 1 + 0j,...
Ceci s'obtient très facilement en multipliant successivement les échantillons de la partie réelle par +1, 0, -1, 0 et en faisant de même avec la partie imaginaire (avec un décalage).
Le spectre obtenu apparaît à la **figure 16b**. En additionnant la partie réelle et la partie imaginaire, on retrouve alors le signal en représentation réelle (voir **figure 16c**).

Conversion avec inversion de spectre

Dans certains cas, le signal peut subir une inversion de spectre en cours de traitement. C'est le cas du récepteur UHF décrit dans l'un des paragraphes précédents. Le HSP50016 offre l'option de retourner à nouveau le spectre pour retrouver la distribution initiale des fréquences. En mode complexe, on y parvient en accordant le NCO sur la bande latérale ad hoc. En mode réel, un bit spécifique du registre de contrôle 4 active la fonction d'inversion spectrale.

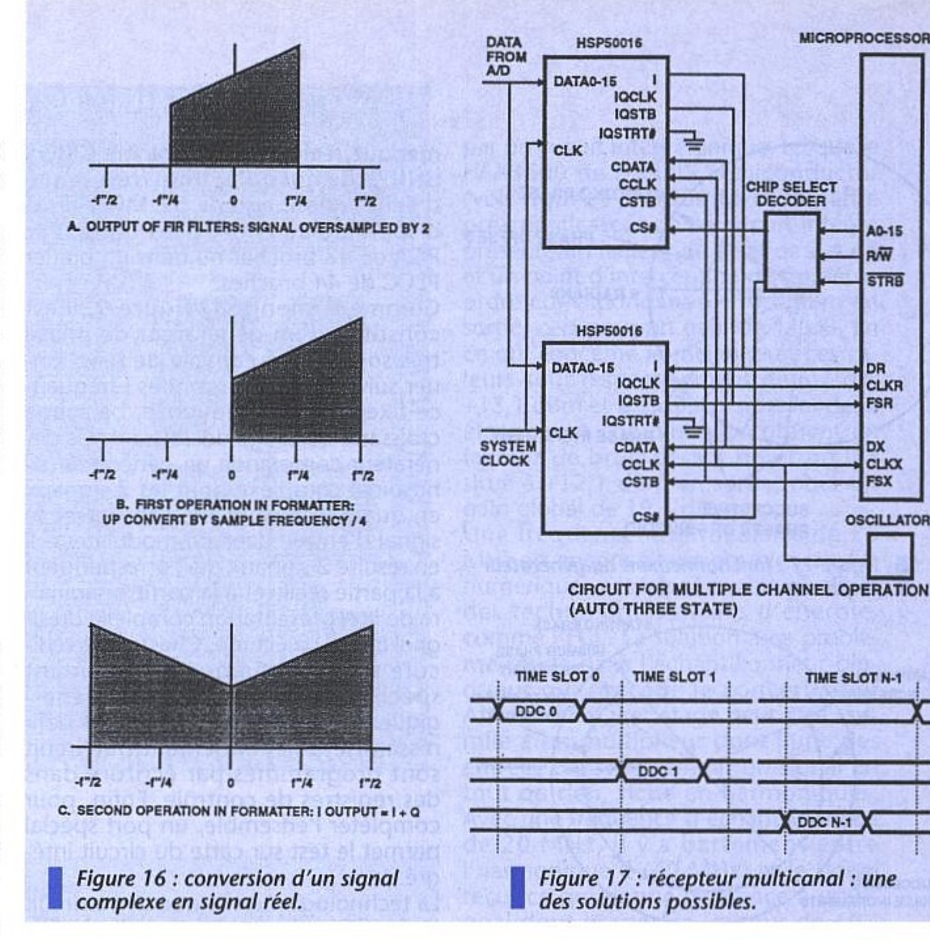


Figure 16 : conversion d'un signal complexe en signal réel.

Figure 17 : récepteur multicanal : une des solutions possibles.

Transfert des données après traitement

Physiquement, le HSP50016 est muni de 2 ports série I et Q compatibles avec la plupart des processeurs numériques (DSP).
Des signaux d'horloge et de synchronisation contrôlent le transfert des données. L'éventail des possibilités offertes à ce niveau est très large.
L'utilisateur peut choisir entre une représentation complexe ou une représentation réelle du signal de sortie. Dans le premier cas, les informations peuvent être disponibles simultanément ou en alternance sur les 2 ports, ou encore en alternance avec utilisation exclusive du port I. Dans le second cas, les données apparaissent sur le port I. Il peut aussi définir le format des données en sortie :
- 16, 24, 32, ou 38 bits
- binaire pur, complément à deux, binaire signé, virgule flottante.
Signalons enfin qu'il est possible de coupler plusieurs DDC ensemble pour les relier à un unique DSP ou microprocesseur.
On peut ainsi réaliser un récepteur multicanal (voir **figure 17**).

Programmation

La programmation du HSP50016 est mémorisée dans 8 registres de contrôle. Ces registres mémorisent des "mots de contrôle" d'une longueur de 40 bits et sont chargés par l'intermédiaire d'un port série. Chaque mot de contrôle est associé à la programmation d'une des fonctions du circuit. Il est ainsi possible de programmer les paramètres du générateur de phase, de définir le format des données en entrée et en sortie, de définir le coefficient de décimation du HDF et les facteurs d'échelle, de sélectionner ou non l'inversion de spectre du signal reçu, de gérer ou non les dépassements de niveau.

Test

Le HSP50016 est muni d'un port de test (TPA : Test Access Port) qui permet de le contrôler même lorsqu'il est en place sur un circuit imprimé. La procédure de test est conforme au standard IEEE 1149.1. Il est possible d'exécuter de cette manière un certain nombre d'instructions, dont par exemple la lecture/écriture des registres de contrôle.

CONCLUSION

Avec l'arrivée du HSP50016, la numérisation des radiorécepteurs marque un nouveau pas. La technologie actuelle n'est certes pas encore utilisable dans tous les cas. A cet égard, le prix de revient d'un système numérique constitue un obstacle majeur à sa généralisation. Malgré tout, elle est particulièrement intéressante pour la réception de signaux à bande très étroite, lorsqu'il faut effectuer des changements de fréquence immédiats (systèmes à évasion de fréquence), ou, dans les stations fixes lorsqu'il faut recevoir simultanément un grand nombre de canaux. N'en doutons pas, les circuits numériques dédiés aux radiocommunications sont promis à bel avenir !

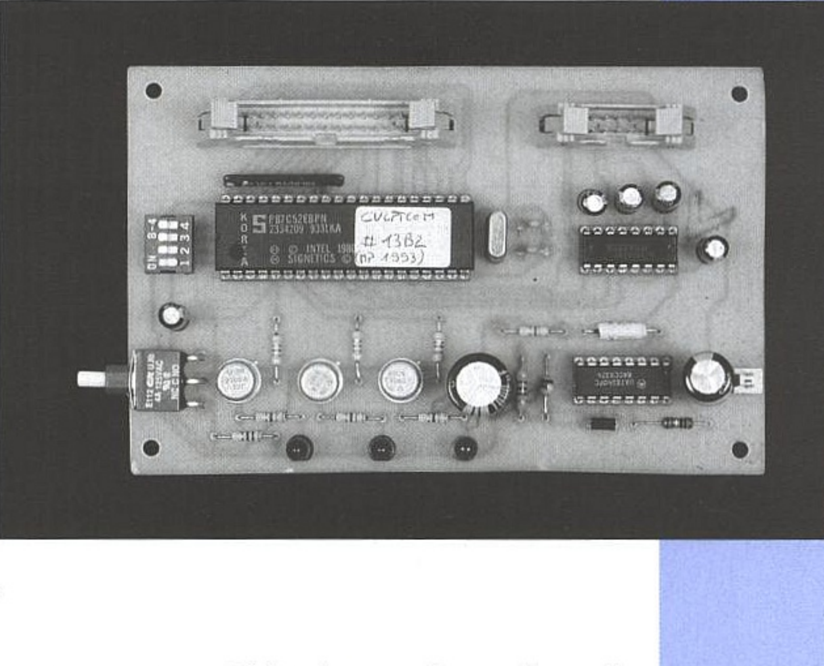
Bibliographie : - DSP Data-book (HARRIS Semiconductor)
- Notice technique du HSP50016
- IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1 - 1990

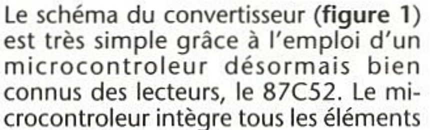
Thierry RIFFLART
HARRIS Semiconductor



CONVERTISSEUR PARALLELE-SERIE REVERSIBLE

Qui n'a jamais eu besoin de connecter un équipement série sur un port parallèle ou vice versa ? Un convertisseur s'impose donc. Mais selon le sens de la conversion il faudra un convertisseur différant. Pour couvrir les deux cas de figure, le convertisseur que nous vous proposons ce mois-ci est réversible.





Le schéma du convertisseur (**figure 1**) est très simple grâce à l'emploi d'un microcontroleur désormais bien connus des lecteurs, le 87C52. Le microcontroleur intègre tous les éléments nécessaires à notre application, à savoir plusieurs ports 8 bits bidirectionnels, un port série, sans oublier une EPROM, une RAM, et des TIMERS. Selon le sens de la conversion, le logiciel du microcontroleur configurera les lignes du port parallèle en entrées ou bien en sorties. On peut donc relier directement les ports du microcontroleur au connecteur du port parallèle.

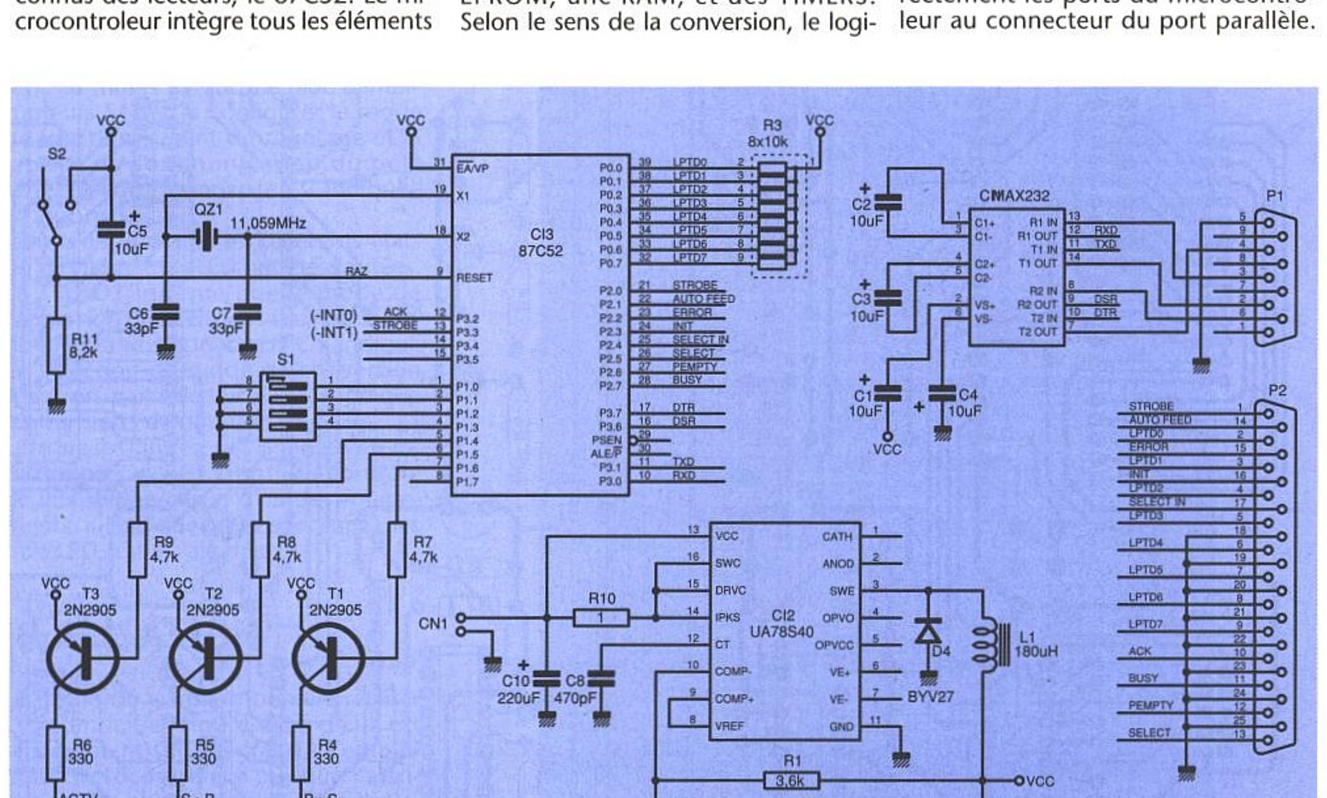


Figure 1 : le schéma



MODE SERIE --> PARALLELE

VERS IMPRIMANTE PARALLELE — VERS PORT SERIE TYPE IBM/PC — DB9 FEMELLE

VERS IMPRIMANTE PARALLELE — VERS PORT SERIE TYPE IBM/PC — DB25 FEMELLE

SENS DU TRANSFERT

SENS DU TRANSFERT

MODE PARALELLE --> SERIE

VERS PORT IMPRIMANTE TYPE IBM/PC — VERS IMPRIMANTE SERIE — DB9 FEMELLE

(Connections à déterminer selon les signaux de controle de flux disponibles sur l'imprimante)

VERS PORT IMPRIMANTE TYPE IBM/PC — VERS IMPRIMANTE SERIE — DB25 FEMELLE

(Connections à déterminer selon les signaux de controle de flux disponibles sur l'imprimante)

SENS DU TRANSFERT

SENS DU TRANSFERT

Figure 2 : les différentes interconnexions

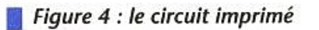


Figure 4 : le circuit imprimé

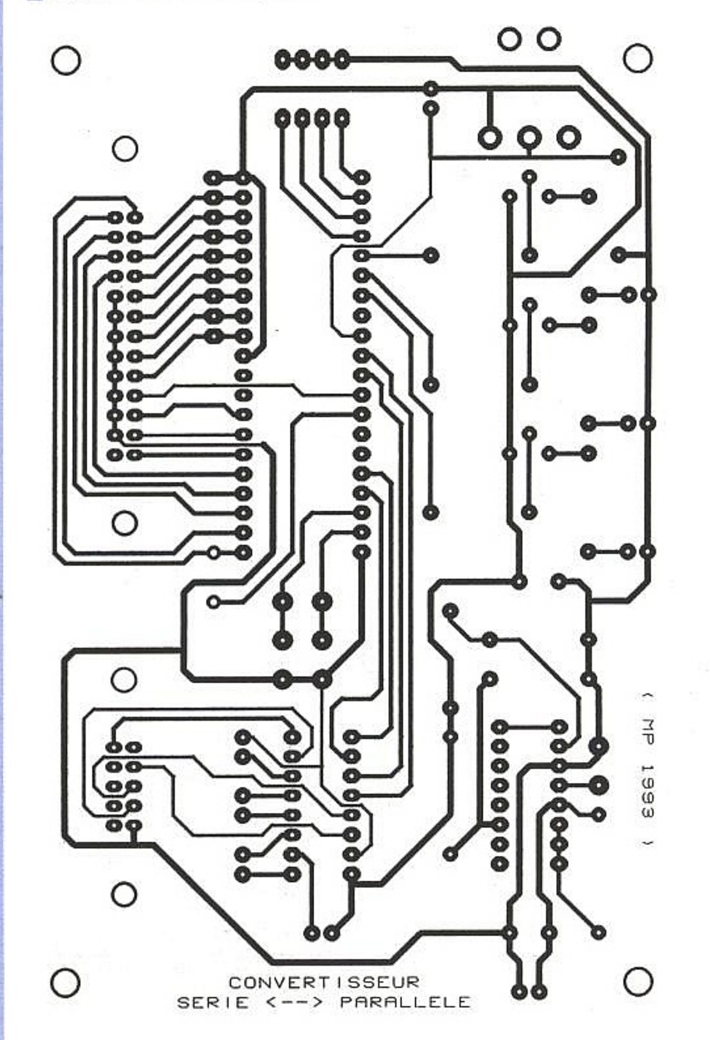
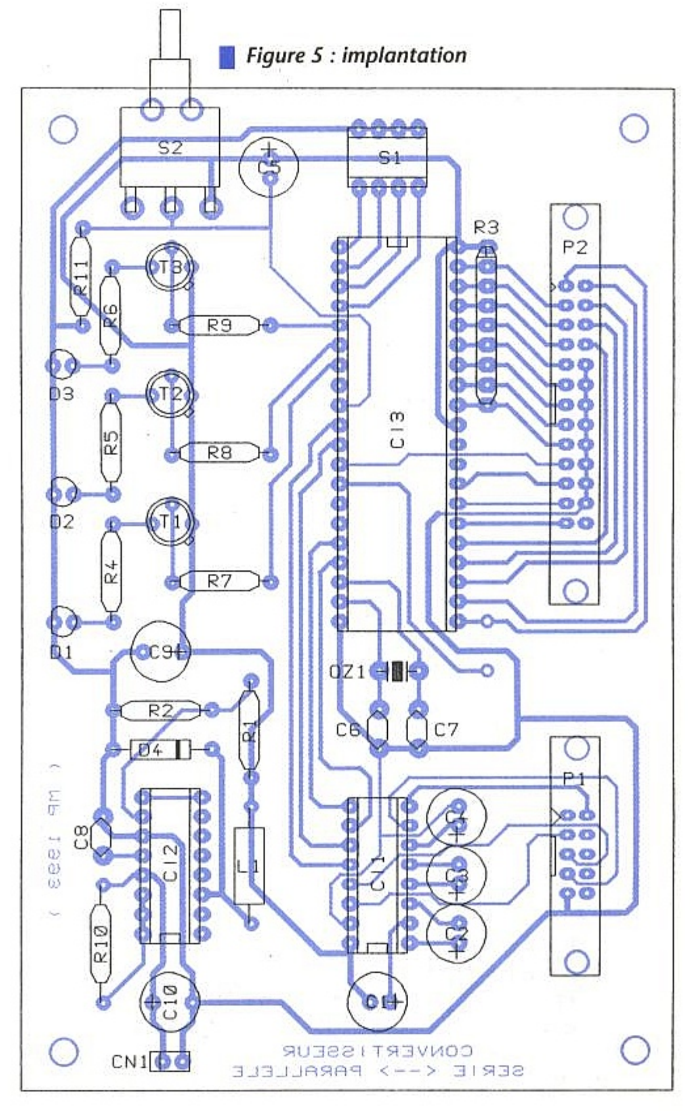
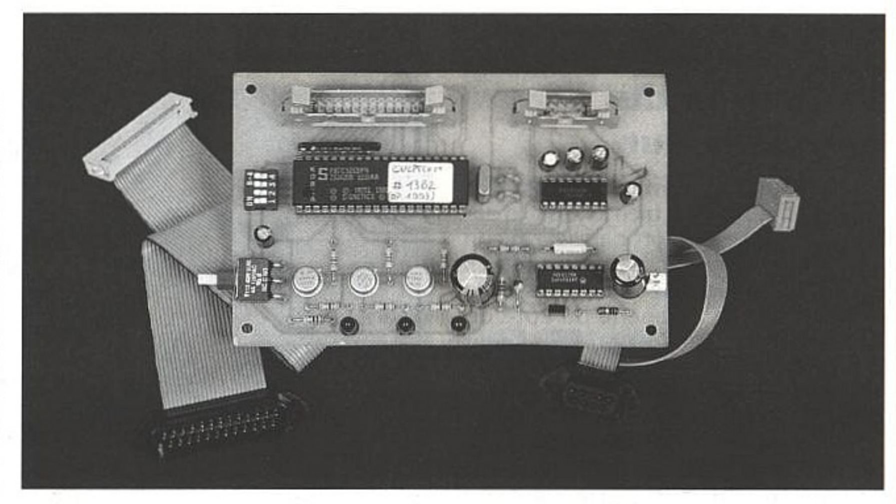


Figure 5 : implantation





Le port P0 du microcontroleur nécessite des résistances de rappels à VCC puisqu'il en est dépourvu. Les signaux ACK et STROBE sont reliés aux entrées INT0 et INT1 afin de permettre une gestion du transfert des données via les interruptions. En ce qui concerne le port série, l'adaptation des niveaux est confiée à un MAX232. Le sens de communication des lignes du port série ne sera donc pas réversible. Pour tenir compte du sens des signaux de contrôle de flux, il faudra donc utiliser un câblage différent selon le sens de la conversion souhaitée. La **figure 2** regroupe les différents câbles possibles.



SW1	SW2	SW3	Vitesse de communication
ON	ON	ON	19200 Bauds
OFF	ON	ON	9600 Bauds
ON	OFF	ON	4800 Bauds
OF	OFF	ON	2400 Bauds
ON	ON	OFF	1200 Bauds
OFF	ON	OFF	600 Bauds
ON	OFF	OFF	300 Bauds
OFF	OFF	OFF	150 Bauds

SW4	Sélection du mode
ON	Conversion Série vers Parallèle
OFF	Conversion Parallèle vers Série

Figure 3 : affectation des switches

L'alimentation de l'ensemble est confiée à un petit régulateur à découpage. De la sorte il est possible d'alimenter le montage par n'importe quel bloc secteur pour calculatrice fournissant au moins 9V DC. Le bloc d'interrupteurs S1 sert à configurer le mode de fonctionnement du montage et la vitesse de communication du port série. La correspondance est indiquée en **figure 3**.
Enfin pour indiquer les différents états du montage trois LED ont été ajoutées. La LED D1 indique que le mode de conversion parallèle vers série est sélectionnée tandis que la LED D2 indique la sélection du mode de conversion série vers parallèle. La LED D3 indique les transferts de données en cours.
Lorsque le transfert des données est interrompu (soit en raison du contrôle de flux soit en raison d'un des équipements qui indique qu'il est occupé), les trois LED restent allumées.

RÉALISATION

Le dessin du circuit imprimé à réaliser est reproduit en **figure 4**. Le circuit est relativement simple grâce à l'emploi d'un microcontroleur puisque l'attribution des ports se fait par une déclaration logicielle.
La vue d'implantation est visible en **figure 5**. N'oubliez pas l'unique strap du montage. Vous noterez que le microcontroleur n'est pas dans le même sens que les autres composants. Donc veillez-y.

Le logiciel

Le listing source du logiciel à implanter dans le microcontroleur ne pouvant tenir dans ces pages, nous vous livrons le code hexadécimal que vous pourrez télécharger par Minitel sur le 3615 code ERP. Listing ci-dessous.
Si vous n'êtes pas équipé pour programmer vous-même le microcontrôleur, vous trouverez des revendeurs de composants tout prêts à vous rendre ce service (si vous achetez le microcontroleur chez eux bien entendu !).

```
:0200000012CD1
:0200030021FBDF
:02000B00216072
:0200130021FECC
:01001B0032B2
:02002300217A40
:10002B00327587807581BFC200C20275113075129F
:10003B0030C201C2047513007589017 58CFC758A79
:10004B0018D2A9D28C759850E5905407F8900224D9
:10005B0093F5CB90021CE893F5CA75C834A2939222
:10006B000330030201A7C295D296D294C2B7D2B67F
:10007B0075A0FFC2A0C2A1C2A37FFF7E0ADFFEDE76
:10008B00FCD2A3D2ACD2AF31002002FB3000F8205F
:10009B0001023137314AF58031410192C296D29536
:1000AB00D294D2B6C2B7D206D2ACD2AAD28BD2BA23
:1000BB00D2BCD2AFC2A7C2A6D2B230A31D3000FAB7
:1000CB003006F730B60AD2A7D202C294C29580EAA4
:1000DB00D295C202314AF59901C5D2A7C20075115A
:1000EB0030751230C294D296D20230A3FDC202C236
:1000FB0096C2A701C530A22430A42120A61E20A79A
:10010B001030020CC202C2B7E510 448FF590D294A6
:10011B00227FFF7EFF30A7E9DEFB DFF7D2B720029D
:10012B000AE590F510748FF590D202227FFF200123
:10013B0004DFFBD20122D2A000000000C2A022C229
:10014B00AFA812E6C0E031F48812E8B51102C20084
:10015B00D0E0D2AF22758CFC758A18D513107513AD
:10016B006420020A200403D29432C294C2043230B7
:10017B000342209903C2983220000BC299C2B2D21B
:10018B00B2C2A7D2063220B61AC0E0C000314AF57F
:10019B0099C299D000D0E0200506C2B2D2B2C2A754
:1001AB00C20532D2A7D202C294C29520B6FDC202BA
:1001BB00D2952191209803C29932C0E0D2D3D204B8
:1001CB00A811E599F6C298D20031F4E8B51204D221
:1001DB00B78011881131F4E8B51204D2B78005202D
:1001EB000202C2B7C2D3D0E03208B8C002783022C4
:1001FB00C20132D2A7C0E0D2D3E580A811F631F408
:10020B008811D204D200E8B51202D205C2D3D0E0D5
:10021B0032ECD8B163C78E1D09FFFFFFFFFEFDFB5C
:10022B00F6284329204D6F72696E2050617363610C
:10023B006C20446563656D62726520313939320FA
:0B024B00526576203A20312E30310041
:00000001FF
```

Code hexa du convertisseur

Utilisation du convertisseur

L'utilisation du convertisseur est très simple, mais il faut bien choisir le câblage. Du coté du port parallèle aucun souci à se faire, un câble direct 25 points suffit quel que soit le sens de la conversion. En revanche du côté du port série, il faudra identifier les lignes qui contrôlent le flux des données.
Dans le cas de la conversion série vers parallèle la ligne qui contrôlera le flux est fournis par le convertisseur, il s'agit de la ligne DTR. Le câblage pour connecter un port série de type IBM/PC est donc sans surprise comme l'indique la figure 2.
Par contre dans le sens parallèle vers série il faudra bien éplucher la documentation de l'imprimante à brancher pour identifier quelle ligne autorise la réception des données pour la relier à l'entrée DSR du convertisseur.
Certaines imprimantes utilisent le signal DTR pour indiquer qu'elles sont prêtes, d'autres non. Par exemple les imprimantes FACIT utilisent le signal SRTS (broche 19 sur un connecteur DB25) pour indiquer qu'elles sont prêtes à recevoir des données.
Si vous n'avez plus la documentation de votre imprimante, l'utilisation d'une boite de jonction vous sera sans aucun doute indispensable pour identifier quelles lignes doivent être raccordées entre elles, pour que tout fonctionne sans perte de données.

P. MORIN

Nomenclature

Résistances

R1 : 3,6 kΩ
R2 : 1,2 kΩ
R3 : Réseau résistif SIL 8x10 kΩ
R4,R5,R6 : 330 Ω 1/2W
R7,R8,R9 : 4,7 kΩ 1/4W
R10 : 1 Ω 1/2W
R11 : 8,2 kΩ 1/4W

Condensateurs

C1 : 220µF / 25V sorties radiales
C2,C3,C4,C5,C10 : 10µF
C6,C7 : 33pF
C8 : 10nF
C9 : 1000µF / 25V sorties radiales

Semi-conducteurs

D1,D2,D3 : Diodes LED
D4 : Diode BYV27
T1,T2,T3 : 2N2905 ou équivalent

Circuits intégrés

IC1 : MAX232
IC2 : UA78S40
IC3 : 87C52

Divers

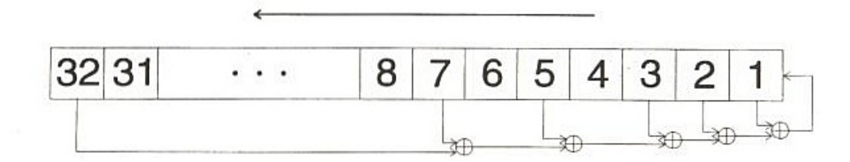
Y1 : Quartz 12 MHz
P1 : Connecteur mâle 10 points série HE10, sorties droites, à souder sur CI (par exemple 3M 2510-5002)
+ connecteur femelle 10 points série HE10,à sertir (par exemple 3M 3473-6000)
+ connecteur femelle DB 9 points à sertir (par exemple Fujitsu FCN-777J009-G/A 9)
+ câble plat 10 conducteurs
P2 : Connecteur mâle 26 points série HE10, sorties droites, à souder sur CI (par exemple 3M2526-5002)
+ connecteur femelle 26 points série HE10, à sertir (par exemple 3M3399-6000)
+ connecteur mâle DB 25 points à sertir (par exemple Fujitsu FCN-777P025-G/A 25)
+ câble plat 25 conducteurs
S1 : Bloc de 4 microswitches en boîtier DIL
S2 : Bouton poussoir à monter coudé sur CI
L1 : Inductance 100nH

GÉNÉRATION DE SÉQUENCES PSEUDO-ALÉATOIRES

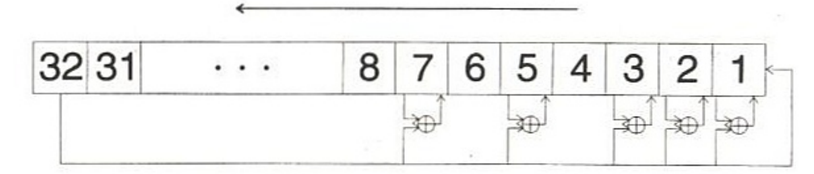
Que ce soit dans des jeux, des procédures de test, les générateurs de bruit ou les transmissions numériques, les générateurs de séquences pseudo-aléatoires sont utilisés chaque fois que l'on désire introduire dans un programme un comportement qui semble être lié au hasard.

Il peut sembler curieux d'obtenir à la suite d'un calcul, un résultat qui soit lié au hasard. En fait le hasard n'intervient pas dans le calcul et celui-ci est parfaitement déterministe, et en effectuant le même calcul avec les mêmes conditions initiales, on obtiendra le même résultat.

Toutefois, pour une personne (ou un processus quelconque) qui ignore de quelle manière le calcul a été effectué, le résultat semblera être aléatoire, c'est pourquoi ces séquences sont appelées pseudo-aléatoires.
Parmi les différentes méthodes utilisés pour générer une séquence pseudo-aléatoire, une méthode très répandue consiste à utiliser des polynômes premiers modulo 2 qui sont définis par une formule du type $x^{11} + x^2 + 1$ que l'on peut abréger en indiquant uniquement les puissances différentes de zéro (11, 2, 0). L'ordre du polynôme (le plus haut degré présenté dans polynôme), ici 11, définit le nombre de séquences différentes qui sont générées avant que celles-ci se répètent. Ce nombre est égal



Première méthode de calcul de la séquence



Deuxième méthode de calcul de la séquence

Les deux méthodes de génération.

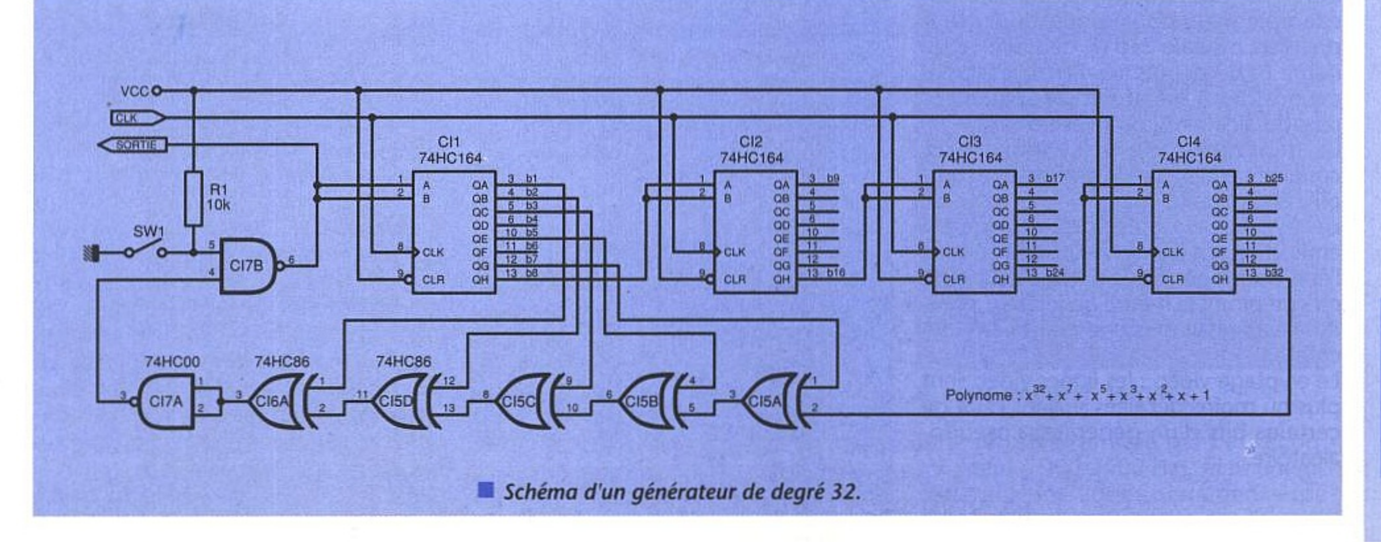


Schéma d'un générateur de degré 32.



à 2^n-1, soit ici 2^{11}-1 = 2047 séquences différentes.
Pour réaliser un circuit qui opère suivant un polynôme donné, il suffit de disposer d'un registre à décalage disposant d'autant de pas que l'ordre du polynôme désiré et un nombre de portes ou-exclusif égal au nombre d'additions dans le polynôme moins un. Sans entrer dans les détails de la théorie, un décalage correspond à une multiplication (ou une division) par deux, et un ou-exclusif à une addition modulo deux. Pour fonctionner correctement, le registre à décalage devra être initialisé à une valeur différente de zéro (bouton Start du schéma), une valeur initiale égale à zéro se perpétuant indéfiniment. Le **schéma 1** réalise un circuit fonctionnant avec le polynôme $x^{32}+x^7+x^5+x^3+x^2+x+1$. (32, 7, 5, 3, 2, 1, 0). Ce pôlynôme a une longueur de séquence de 2^{32}-1 = 4294967295. Les bits sont numérotés de 1 pour le bit généré le plus récent à n pour le bit généré n pas avant.
Cette méthode se prête bien à une réalisation hardware mais est difficilement programmable (ou du moins pas très efficacement) car les processeurs sont généralement pauvres en instructions travaillant au niveau du bit. Une autre méthode existe qui permet d'effectuer tous les ou-exclusifs en une seule passe. Ceci est plus appropriée pour générer des séquences pseudo-aléatoires par programmation. Toutefois, cette deuxième méthode, bien que basée sur le même polynôme que précédemment, **ne génère pas** la même séquence pseudo-aléatoire.
La relation suivante résume l'opération réalisée à chaque pas d'horloge :
b0 = b32, b0 est l'entrée du registre à décalage
b7 = b7^b32
b5 = b5^b32
b3 = b3^b32
b2 = b2^B32
b1 = b1^b32
Le graphique donné en début d'article symbolise les opérations effectuées pour les deux méthodes et le listing 1 réalise cette fonction par programme pour microcontrôleur de type 80C51.
Nous l'avons vu, l'ordre du polynôme détermine la longueur de la séquence générée avant répétition. Le **tableau 1** donne quelques polynômes premiers qui permettent de réaliser des générateurs pseudo-aléatoires. Cette énumération n'est pas exhaustive, il existe par exemple deux polynômes de degré 4 premiers modulo 2, 6 de degré 5, 16 de degré 8, 60 degrés 10. Parmi les applications des générateurs de séquences pseudo-aléatoires, on relèvera :
La génération de bruit blanc : on connectera la sortie du registre à un amplificateur audio.
Le cryptage de message : chaque bit émis et reçu est inversé ou non suivant l'état du bit généré ; si l'émetteur et le récepteur ont le même générateur pseudo-aléatoire, le message pourra être décrypté.
Le cryptage vidéo : les lignes vidéo sont plus ou moins décalées suivant l'état de certains bits d'un générateur pseudo-aléatoire.

J.-L. VERN

```
;-----------------------------------------------------------------------------
; generateur pseudo aleatoire avec un 8051 (generateur de bruit blanc)
; pour generer un bruit blanc, connecter la ligne P1.0 a un amplificateur
;-----------------------------------------------------------------------------
; definition des noms des segments de donnees
SEGMENT_DATA    SEGMENT DATA
SEGMENT_STACK   SEGMENT IDATA    ; le stack est en idata

        RSEG    SEGMENT_DATA
ALEA:   DS      4       ; sequence pseudo aleatoire
        RSEG    SEGMENT_STACK
        DS      20              ; un peu de place pour le stack
; vecteurs de reset
        CSEG    AT      0       ; vecteur de reset en 0
        AJMP    START

SEQUEN: MOV     A,ALEA+0        ; MSB
        MOV     C,ACC.7         ; prend le MSB du MSB (bit 32)
        JC      SEQUE1          ; teste si bit 32 == 1
; le jmp suivant sert a egaliser les temps
        JMP     DECALE          ; 0, decale un 0 sans faire de ou-exclusif
                                ; (0 ou-ex X = X)
; bit 32 a 1, fait un ou exclusif sur les bits 7, 5, 3, 2, 1
SEQUE1: XRL     ALEA+3,#01010111B
; ici, le ou exclusif ne s'effectue que sur le seul LSB. Avec un autre polynome
; il aurrait pu en etre autrement...

; decale d'un bit vers la gauche en poussant la retenue dans le LSB
; le bit de poids le plus fort est ici le bit 32, le poids faible le bit 1
; (et non pas 0 si on suit la notation habituelle)
DECALE: MOV     A,ALEA+3        ; LSB
        RLC     A               ; decale a gauche
        MOV     ALEA+3,A
        MOV     A,ALEA+2
        RLC     A               ; propage la retenue dans le registre
        MOV     ALEA+2,A
        MOV     A,ALEA+1
        RLC     A
        MOV     ALEA+1,A
        MOV     A,ALEA+0        ; jusqu'au MSB
        RLC     A
        MOV     ALEA+0,A
        RET
;-----------------------------------------------------------------------------
; debut du programme
;-----------------------------------------------------------------------------
START:  MOV     SP,#SEGMENT_STACK-1     ; stack : premiere place libre
        MOV     ALEA+3,#0FFH            ; initialise le registre avec une
                                        ; valeur non nulle
START0: CALL    SEQUEN                  ; genere une nouvelle valeur
        MOV     P1.0,C                  ; sort le bit calcule sur P1.0
        JMP     START0                  ; et boucle
; boucle en 25 cycles. Avec un Xtal a 12MHz, la sequence est reproduite tous
; les 12 jours.
        END
```

Listing 1

(1, 0)	(26, 6, 2, 1, 0)	(51, 6, 3, 1, 0)	(76, 5, 4, 2, 0)
(2, 1, 0)	(27, 5, 2, 1, 0)	(52, 3, 0)	(77, 6, 5, 2, 0)
(3, 1, 0)	(28, 3, 0)	(53, 6, 2, 1, 0)	(78, 7, 2, 1, 0)
(4, 1, 0)	(29, 2, 0)	(54, 6, 5, 4, 3, 2, 0)	(79, 4, 3, 2, 0)
(5, 2, 0)	(30, 6, 4, 1, 0)	(55, 6, 2, 1, 0)	(80, 7, 5, 3, 2, 1, 0)
(6, 1, 0)	(31, 3, 0)	(56, 7, 4, 2, 0)	(81, 4, 0)
(7, 1, 0)	(32, 7, 5, 3, 2, 1, 0)	(57, 5, 3, 2, 0)	(82, 8, 7, 6, 4, 1, 0)
(8, 4, 3, 2, 0)	(33, 6, 4, 1, 0)	(58, 6, 5, 1, 0)	(83, 7, 4, 2, 0)
(9, 4, 0)	(34, 7, 6, 5, 2, 1, 0)	(59, 6, 5, 4, 3, 1, 0)	(84, 8, 7, 5, 3, 1, 0)
(10, 3, 0)	(35, 2, 0)	(60, 1, 0)	(85, 8, 2, 1, 0)
(11, 2, 0)	(36, 6, 5, 4, 2, 1, 0)	(61, 5, 2, 1, 0)	(86, 6, 5, 2, 0)
(12, 6, 4, 1, 0)	(37, 5, 4, 3, 2, 1, 0)	(62, 6, 5, 3, 0)	(87, 7, 5, 1, 0)
(13, 4, 3, 1, 0)	(38, 6, 5, 1, 0)	(63, 1, 0)	(88, 8, 5, 4, 3, 1, 0)
(14, 5, 3, 1, 0)	(39, 4, 0)	(64, 4, 3, 1, 0)	(89, 6, 5, 3, 0)
(15, 1, 0)	(40, 5, 4, 3, 0)	(65, 4, 3, 1, 0)	(90, 5, 3, 2, 0)
(16, 5, 3, 2, 0)	(41, 3, 0)	(66, 8, 6, 5, 3, 2, 0)	(91, 7, 6, 5, 3, 2, 0)
(17, 3, 0)	(42, 5, 4, 3, 2, 1, 0)	(67, 5, 2, 1, 0)	(92, 6, 5, 2, 0)
(18, 5, 2, 1, 0)	(43, 6, 4, 3, 0)	(68, 7, 5, 1, 0)	(93, 2, 0)
(19, 5, 2, 1, 0)	(44, 6, 5, 2, 0)	(69, 6, 5, 2, 0)	(94, 6, 5, 1, 0)
(20, 3, 0)	(45, 4, 3, 1, 0)	(70, 5, 3, 1, 0)	(95, 6, 5, 4, 2, 1, 0)
(21, 2, 0)	(46, 8, 5, 3, 2, 1, 0)	(71, 5, 3, 1, 0)	(96, 7, 6, 4, 3, 2, 0)
(22, 1, 0)	(47, 5, 0)	(72, 6, 4, 3, 2, 1, 0)	(97, 6, 0)
(23, 5, 0)	(48, 7, 5, 4, 2, 1, 0)	(73, 4, 3, 2, 0)	(98, 7, 4, 3, 2, 1, 0)
(24, 4, 3, 1, 0)	(49, 6, 5, 4, 0)	(74, 7, 4, 3, 0)	(99, 7, 5, 4, 0)
(25, 3, 0)	(50, 4, 3, 2, 0)	(75, 6, 3, 1, 0)	(100, 8, 7, 2, 0)

Tableau 1



MODULES DE TRANSMISSION VIDÉO SUR PAIRE TORSADÉE

Comme il est loin, le temps où on ne savait guère transmettre sur une «paire torsadée» qu'une simple voie téléphonique 300-3400 Hz!

Les données numériques y circulent maintenant à plusieurs mégabits par seconde, tandis que d'intéressantes applications sont en vue dans le domaine de la vidéo.

Sans chercher à atteindre une qualité «broadcast», nous allons voir que des circuits fort simples peuvent permettre de transmettre une image très honorable à des centaines de mètres, voire plus.

Cela sur du simple câble bifilaire pour systèmes d'alarme, ou sur une paire d'un câble d'installation téléphonique ou de sonnerie: les applications sont évidentes en matière de surveillance et de contrôle d'accès...



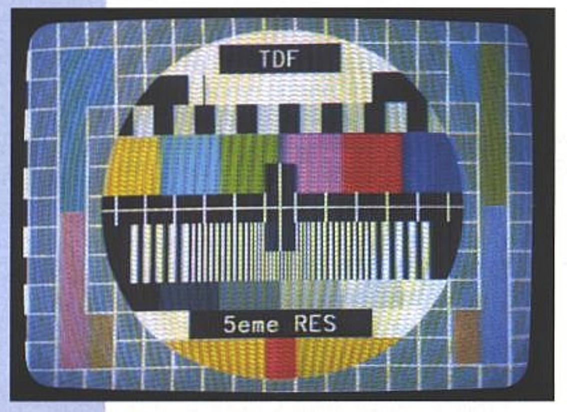
DEUX CARTES D'EVALUATION

Il est bien évident que le schéma exact à utiliser dépendra toujours plus ou moins de l'application envisagée et des contraintes qui en découlent.
Il est cependant possible de construire très facilement deux petits modules (un émetteur et un récepteur) permettant de se faire une idée des performances pouvant être obtenues sur le terrain.
Ces deux cartes d'évaluation suffiront par exemple pour raccorder une caméra à un moniteur à l'aide de quelques centaines de mètres de câble téléphonique, ou de ce petit scindex utilisé en alarme et en pyrotechnie.
Utilisé entre deux pièces d'une habitation ou entre deux appartements d'un immeuble, voire même entre deux bâtiments, il pourra tout aussi bien servir à retransmettre des programmes lus par un magnétoscope ou reçus par voie hertzienne ou par câble (rien n'interdit de se servir d'une seconde paire pour le son!).
Naturellement, à partir d'une certaine longueur de ligne, il faut s'attendre à une dégradation de plus en plus sensible des images couleur. Mais en noir-et-blanc sur un écran de petite taille, il ne serait pas ridicule d'espérer mettre en place une surveillance à un ou deux kilomètres de distance, et peut-être davantage!
Même si cela va sans dire, précisons tout de même que l'opération ne peut en aucune façon s'envisager sur une ligne téléphonique d'un réseau commuté public ou privé (limitée par l'autocommutateur à 300-3400 Hz), mais uniquement sur une ligne totalement «spécialisée».



■ *Le jeu de circuits MAXIM.*





Mire retransmise sans préaccentuation.

Un problème d'adaptation

Support de transmission analogique «large bande» par excellence, le câble coaxial est le type même de la ligne «asymétrique» blindée.
Par définition, une «paire torsadée» est à l'inverse une ligne «symétrique», qui n'a souvent nul besoin de blindage si on respecte bien sa symétrie au niveau des équipements d'extrémité qu'on vient y raccorder.
Songeons aux rubans bifilaires «300Ω» pour antennes, rarement blindés mais capables de transporter sans encombre de très faibles signaux à des centaines de MHz, sur des dizaines de mètres s'il le faut.
Pour relier deux équipements vidéo ordinaires par une ligne symétrique, il va naturellement falloir opérer une conversion asymétrique-symétrique à une extrémité, et une conversion symétrique-asymétrique à l'autre.
Une adaptation d'impédance s'imposera également entre les 75Ω classiques en vidéo et les 100 à 600Ω des paires torsadées usuelles.
Enfin, une amplification et éventuellement une égalisation pourront être nécessaires pour compenser l'atténuation introduite par la ligne sur les fréquences les plus élevées, surtout en cas de grande longueur (1000 mètres et plus).

Une application des amplificateurs «à transconductance»

Particulièrement appréciés en vidéo, les amplificateurs dits «à transconductance» ressemblent aussi peu que possible aux classiques amplis-op.
Pour commencer, ils ne sont pas contre-réactionnés, ce qui élimine d'emblée un risque potentiel d'instabilité.
Ensuite, leur bande passante est exceptionnellement élevée (de l'ordre de 250 MHz et plus), même à des gains déjà significatifs.
Mais leur structure très particulière permet tout de même de modeler à volonté leur fonction de transfert.
C'est à MAXIM que l'on doit deux références particulièrement innovantes d'amplis à transconductance: le MAX435 à entrée et sortie différentielles, et le MAX436 à entrée différentielle et à sortie unipolaire.
Le schéma de base de la **figure 1** montre comment est fixé le gain en boucle ouverte: c'est le produit d'un gain en courant interne «K» avec le rapport entre l'impédance de sortie ZL et celle du «réseau de transconductance» ZT monté par l'utilisateur entre les broches Z+ et Z-.
Un simple rapport de résistances conduira à un gain précis, tandis que l'introduction d'un réseau RC ou LC (voire même d'un quartz) entre Z+ et Z- ouvrira de larges perspectives de modelage de la courbe de réponse (réalisation de filtres).
Le schéma interne simplifié du MAX435 (**figure 2**) montre les solutions techniques adoptées pour obtenir ce comportement. On remarquera que cet ampli est muni de véritables entrées différentielles à haute impédance (environ 800 kΩ) et de sorties également différentielles fonctionnant en sources de courant, elles aussi à haute impédance (environ 3,2 kΩ) : c'est le dipôle de charge qui fixera à lui seul l'impédance de sortie.
On le voit, nous sommes décidément très loin de l'amplificateur opérationnel classique!
Le MAX436 diffère du MAX435 par le fait qu'il est muni d'une seule sortie, et que son rapport interne «K» est de 8 au lieu de 4.
Ces deux composants sont prévus pour être alimentés en plus et moins 5V, et possèdent une broche permettant de régler, dans une certaine mesure, les courants des générateurs internes: on peut ainsi privilégier ou bien une faible dissipation, ou alors un courant de sortie élevé.
Cette même broche Rset permet également de mettre l'amplificateur en «stand-by», mode dans lequel sa

VO1 = K (Z1/Zt) VIN
VO2 = K (Z2/Zt) VIN
(k : 4 ± 2,5% pour le MAX435
8 ± 2,5% pour le MAX436)

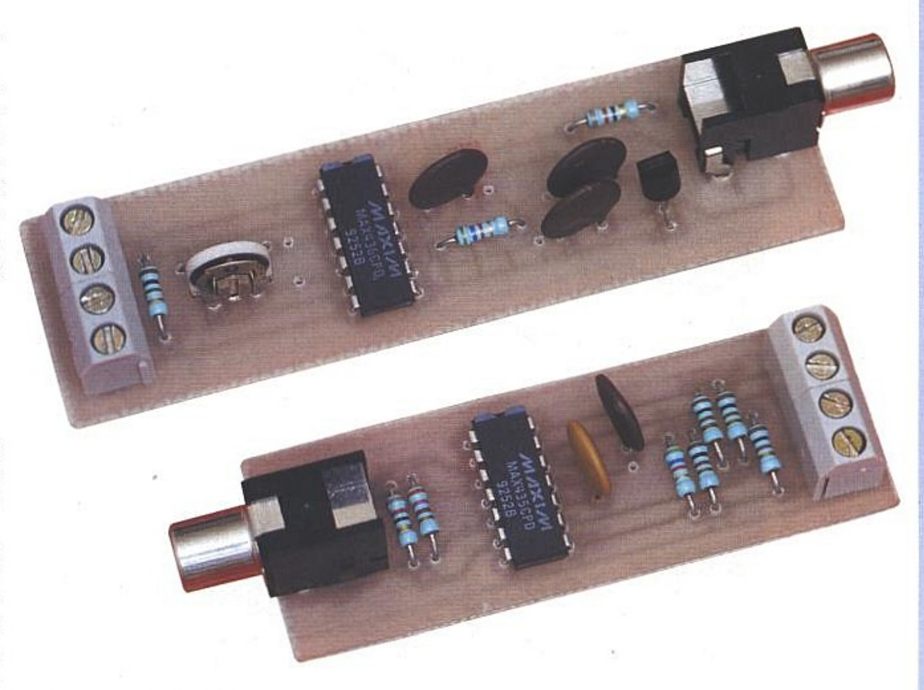
Figure 1 : schéma de base d'application

Figure 2 : structure interne

R4 à R7 : 1/2W SFR25
C1 à C6 : céramiques 25V

ATTENTION ! : Les alimentations de ces modules ne doivent pas être communes avec celles des équipements vidéo qui y sont racordés

consommation tombe à quelques centaines de µA.
On ne manquera pas d'apprécier la réjection de mode commun tout à fait exceptionnelle offerte par ces composants: 53 dB à 10 MHz! Dans notre cas, cela signifie que l'on pourra utiliser sans risque des paires torsadées dépourvues de blindage, et cela même si elles appartiennent à un câble dans lequel circule déjà tout autre chose que de la vidéo.



Le module émetteur

Sur le schéma général de la **figure 3**, le module émetteur se situe à gauche comme en témoigne son connecteur «VIDEO IN».
L'impédance de cette entrée est fixée par une résistance de 75Ω (mais on peut se contenter de 82Ω sans grand inconvénient), tandis que le gain de l'ampli est déterminé par une simple résistance de 220Ω : il n'y a donc pas de «préaccentuation», mais on pourrait y songer en cas de ligne très longue (il suffirait d'ajouter un réseau RC en parallèle).
Chacune des deux sorties (différentielles) de l'ampli est munie d'une résistance de charge de 50Ω constituée par deux éléments de 100Ω reconstituant une «masse virtuelle».
Cela permet d'alimenter le circuit sous une tension unique de 10 à 12V, en l'occurence un petit bloc secteur «prise de courant» à sortie régulée ou des piles, la consommation étant de l'ordre de 200 mA.
Cette disposition présente deux inconvénients mineurs: une dissipation importante des résistances de sortie (près de 500 mW pour chacune, il est donc normal qu'elles chauffent franchement), et une séparation impérative entre la masse de la source vidéo et l'alimentation.
Il nous a toutefois paru préférable de ne pas nous embarrasser d'une alimentation symétrique, puisque l'entrée différentielle de l'ampli nous permettait ce petit tour de passe-passe!
Nous partons donc du principe que l'impédance caractéristique de la ligne (Zp) est de 100Ω, ce qui est en principe assez proche de la réalité. Il serait de toute façon facile de fixer une valeur différente en modifiant simplement quatre résistances.

Le module récepteur

Côté réception, la ligne est donc bouclée sur une résistance de 100Ω afin de respecter l'impédance caractéristique de l'ensemble.
Le cas échéant, si plusieurs récepteurs devaient fonctionner sur la même paire, on pourrait songer à reporter cette «terminaison» une fois pour toutes à l'extrémité de la ligne.
C'est là que l'entrée différentielle du MAX436 donne sa pleine mesure, en annulant de façon quasiment parfaite tout bruit de mode commun collecté par la paire.
Dans la plupart des cas, le réseau de transconductance peut se limiter à une résistance ajustable, servant à fixer le gain en fonction de l'atténuation introduite par la ligne, de façon à recueillir 1V de vidéo en sortie.
En cas de transmission d'images couleur sur une ligne particulièrement longue, on pourra introduire une égalisation en ajoutant un réseau RC ou un condensateur de quelques dizaines de pF.
La sortie (unique cette fois-ci) de l'ampli étant auto-polarisée à Vcc/2, une masse virtuelle de qualité a été créée à l'aide du composant spécialisé TLE2426 de TEXAS INSTRUMENTS.
Comparable à un régulateur «3 pattes», ce composant produit une tension exactement égale à la moitié de l'alimentation, et peut fournir ou absorber une vingtaine de mA.
L'impédance de sortie du module est fixée par une simple résistance de 75Ω, pouvant là encore être remplacée par une 82Ω sans incidence décelable sur la qualité des résultats.
Comme à l'émission, l'alimentation se fera sous 10V mais avec une consommation nettement inférieure (moins de 50 mA).
Il est important de bien respecter cette valeur: au delà de 12V la vie des amplis est menacée, tandis qu'en dessous de 9V le fonctionnement n'est plus assuré: seule une pile 9V pratiquement neuve peut convenir, ou bien un accu parfaitement chargé: à réserver pour les essais, et encore!

MISE EN OEUVRE PRATIQUE

Les deux modules peuvent être câblés chacun sur un étroit circuit imprimé

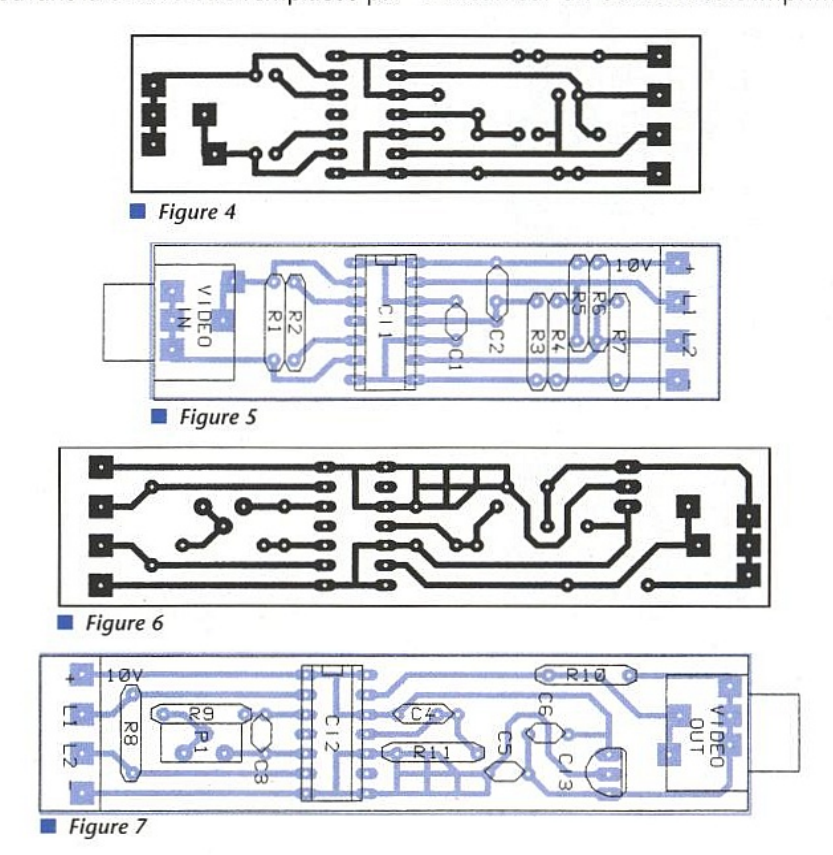


Figure 4

Figure 5

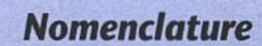
Figure 6

Figure 7



muni d'une embase RCA (pour la vidéo 1V, 75Ω) et d'un bornier à vis pour l'alimentation et la paire torsadée.
La **figure 4** fournit le tracé du cuivre et la **figure 5** le plan d'implantation de l'émetteur, les **figures 6 et 7** concernant pour leur part la carte de réception.
Lors de la mise en service, il importe de veiller à ne pas «croiser» les fils de la paire torsadée: L1 de l'émetteur doit bien aller à L1 du récepteur et réciproquement pour L2. La plupart du temps, des couleurs différentes permettent d'éviter toute confusion, mais la vigilance est de rigueur s'il faut abouter plusieurs tronçons.
En présence de câbles multiconducteurs, on veillera à bien utiliser deux fils d'une même paire, c'est à dire torsadés ensemble: cela se voit nettement au dénudage.
Insistons encore une fois sur la nécessité d'employer des alimentations de bonne qualité (régulées entre 10 et 12V), et totalement indépendantes: nul point commun ne doit exister avec l'équipement vidéo raccordé (et notamment sa masse), ni entre les alimentations de l'émetteur et du récepteur.
Aucun réglage n'est à prévoir à l'émission, mais on devra ajuster le gain de l'ampli de réception en fonction des caractéristiques de la ligne. Cela se fera tout simplement par examen de l'image reçue, que l'on cherchera à rendre aussi bonne que possible en combinant le réglage du module avec ceux du moniteur.
Un gain excessif mène à une saturation, surtout dans les hautes lumières, tandis qu'un gain insuffisant se traduit par une perte de contraste, voire de synchronisation.
Bien entendu, un contrôle oscilloscopique peut aussi être effectué, notamment si l'image reçue ne donne pas entière satisfaction.
La plupart du temps, toutefois, on risquera d'être fort agréablement surpris par la qualité obtenue.
Mais nous laisserons à nos lecteurs le soin d'en juger de leurs propres yeux!

Patrick GUEULLE



Nomenclature

Résistances

R2, P1 : 220 Ω
R1, R10 : 75 ou 82 Ω
R3, R11 : 5,6 kΩ
R4, R5, R6, R7, R8, R9 : 100 Ω

Condensateurs

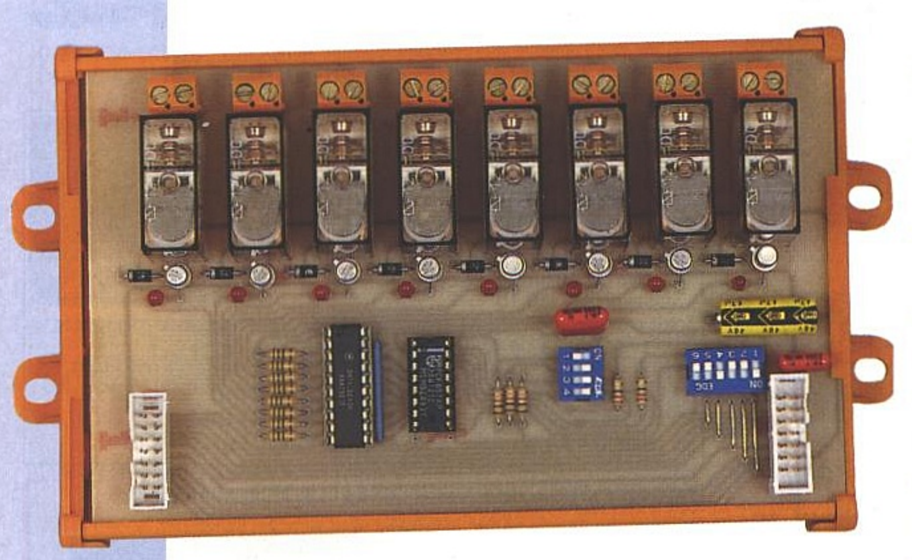
C2, C4, C5, C6 : 100nF
C3 : optionnel

Circuits intégrés

CI1 : MAX435
CI2 : MAX436
CI3 : TLE2426



DOMESTICUS : BORNIER DE SORTIES 8 VOIES ET ALIMENTATION



Voici la deuxième partie de notre réalisation Domotique : Les borniers de sorties Digitales et d'alimentation des voies d'entrées-sorties. Ces deux borniers, reliés au bornier d'entrées Digitales décrit le mois dernier, permettent de réaliser un module d'entrées/sorties digitales complet prêt à être raccordé sur votre unité centrale.

Description du bornier 8 sorties digitales

Schéma électrique

Le schéma éléctrique du bornier 8 sorties digitales est donné **figure 1**. Comme pour le bornier d'entrées digitales, il utilise le circuit spécialisé PHILIPS COMPOSANTS PCF 8574. La liaison vers le réseau I2C est assurée par les connecteurs HE10 14 points K1 et K2. Le réseau utilisé est sélectionné par l'intermédiaire du Dip-switch S2, et l'adresse du circuit PCF 8574 sur ce réseau par S1 (commande des trois broches d'adresses A0, A1 et A2). Le **tableau 1** indique le positionnement des switches de S2 en fonction du réseau à sélectionner. Les huits sorties du PCF 8574 sont inversées par l'octuple tampon inverseur 74LS240 ; une résistance de 330 Ω maintient un niveau haut à l'entrée de l'inverseur en cas de non activation de la sortie du PCF 8574. Chaque sortie de tampon inverseur pilote un transistor 2N2222 au travers d'une LED et d'une résistance de limitation du courant. La LED possède deux fonctions : la visualisation de l'état de la voie, et le blocage du transistor pour un niveau logique bas en sortie de la porte (tension de sortie entre 0 et 0,3V). Chaque transistor commande en 12V un relais (sorties contacts secs, prévues pour 250V 10A) dont les contacts Travail et Commun sont accessibles par les borniers B1 à B8.

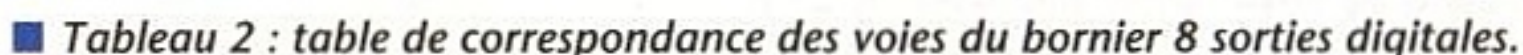
Implantation du bornier

L'implantation du bornier qui est donnée **figure 2** reprend les principes utilisés pour ces modules d'entrées-sorties. On retrouve les connecteurs de réseaux I2C (K1 et K2) dans les coins

Réseau	DIP-SWITCH S2					
	1	2	3	4	5	6
1	ON	OFF	OFF	ON	OFF	OFF
2	OFF	ON	OFF	OFF	ON	OFF
3	OFF	OFF	ON	OFF	OFF	ON

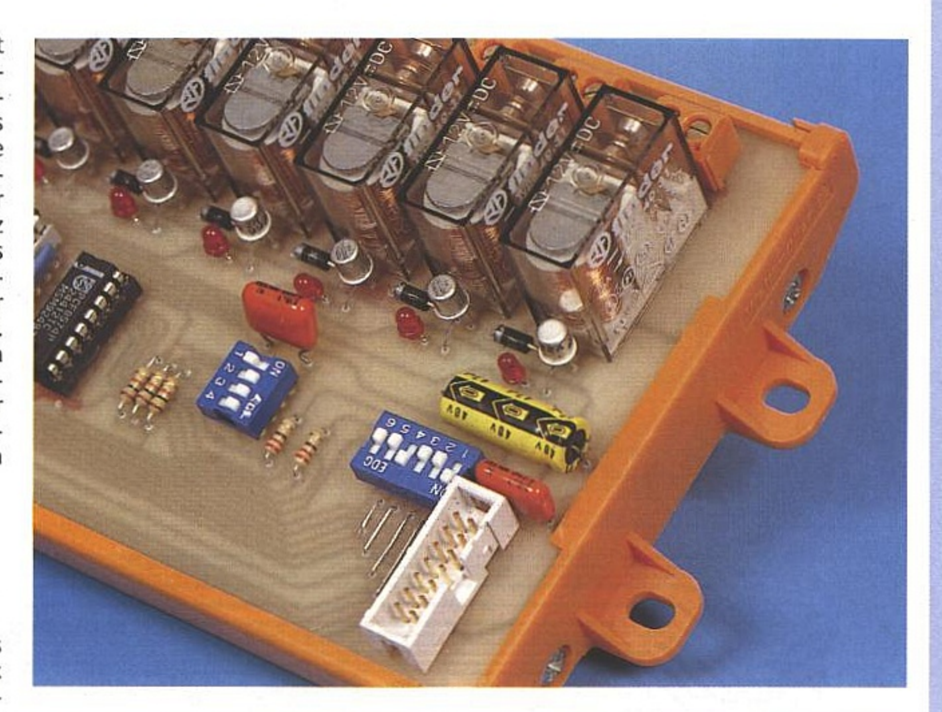
■ *Tableau 1 : positionnement des switches de S2 en fonction du réseau à sélectionner.*

IC1 PCF 8574			IC2 74SL240		Activation Relais	Bornier
octet à transmettre	bit	broche	broche entrée	broche sortie		
xxxx xxx0	P0	4	17	3	Rel2	B2
xxxx xx0x	P1	5	4	16	Rel3	B3
xxxx x0xx	P2	6	6	14	Rel5	B5
xxxx 0xxx	P3	7	8	12	Rel7	B7
xxx0 xxxx	P4	9	11	9	Rel8	B8
xx0x xxxx	P5	10	13	7	Rel6	B6
x0xx xxxx	P6	11	15	5	Rel4	B4
0xxx xxxx	P7	12	2	18	Rel1	B1


■ *Tableau 2 : table de correspondance des voies du bornier 8 sorties digitales.*



gauche et droite en haut du circuit, et les borniers de raccordement de sorties en bas. Pour des raisons de simplification de l'implantation, nous n'avons pas respecté l'ordre logique entre les bits de l'octet de sortie du circuit PCF8574, et le numéro du bornier de sortie contacts secs. Si vous désirez écrire vos propres programmes, il vous faudra utiliser la table de correspondance du **tableau 2** pour commander chaque voie de sortie de ce bornier. Pour les lecteurs créant leur applicatif à l'aide de l'Atelier Logiciel de DOMESTICUS, ce problème ne se posera pas car la table de correspondance a été intégrée dans le driver de communication de ces borniers.



Description du bornier alimentation

Le bornier alimentation des modules d'entrées-sorties permet de fournir aux différents borniers raccordés une tension de +5V (500mA) et +12V (2A). On a donc la possibilité de raccorder jusqu'à 4 borniers de sorties Digitales (consommation de 450mA en 12V si tous les relais sont activés). La consommation sur le +5V reste inférieure à 10mA pour chaque bornier d'entrées ou de sorties digitales. De plus, ce bornier permet le raccordement des réseaux I2C avec l'installation Domotique de votre habitation, via un connecteur à vis, ainsi que le branchement de l'horloge à afficheurs géants que nous publierons dans un prochain numéro. Le brochage de ce bornier à vis est représenté **figure 3**.

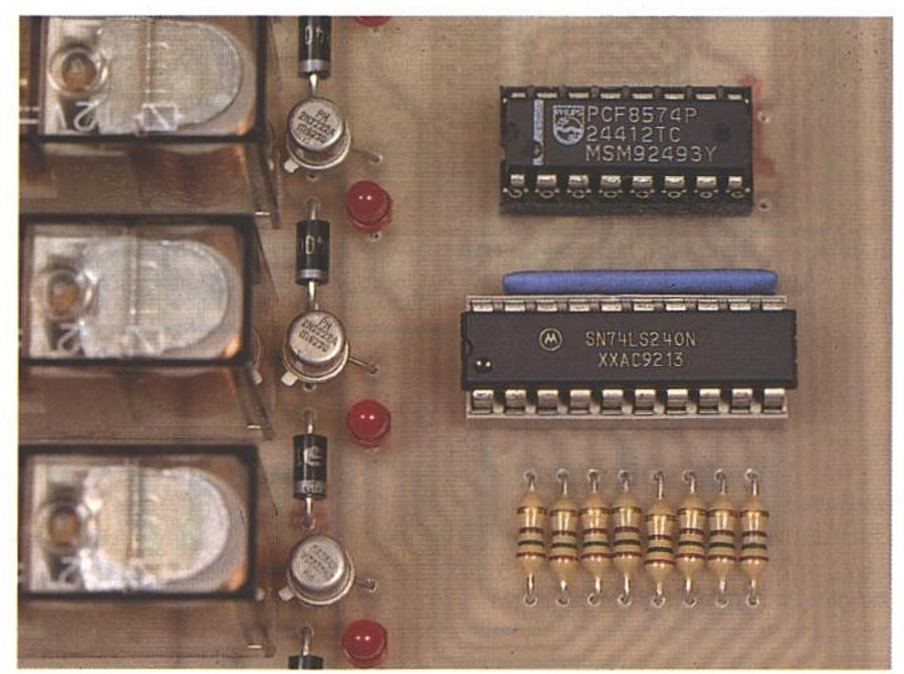
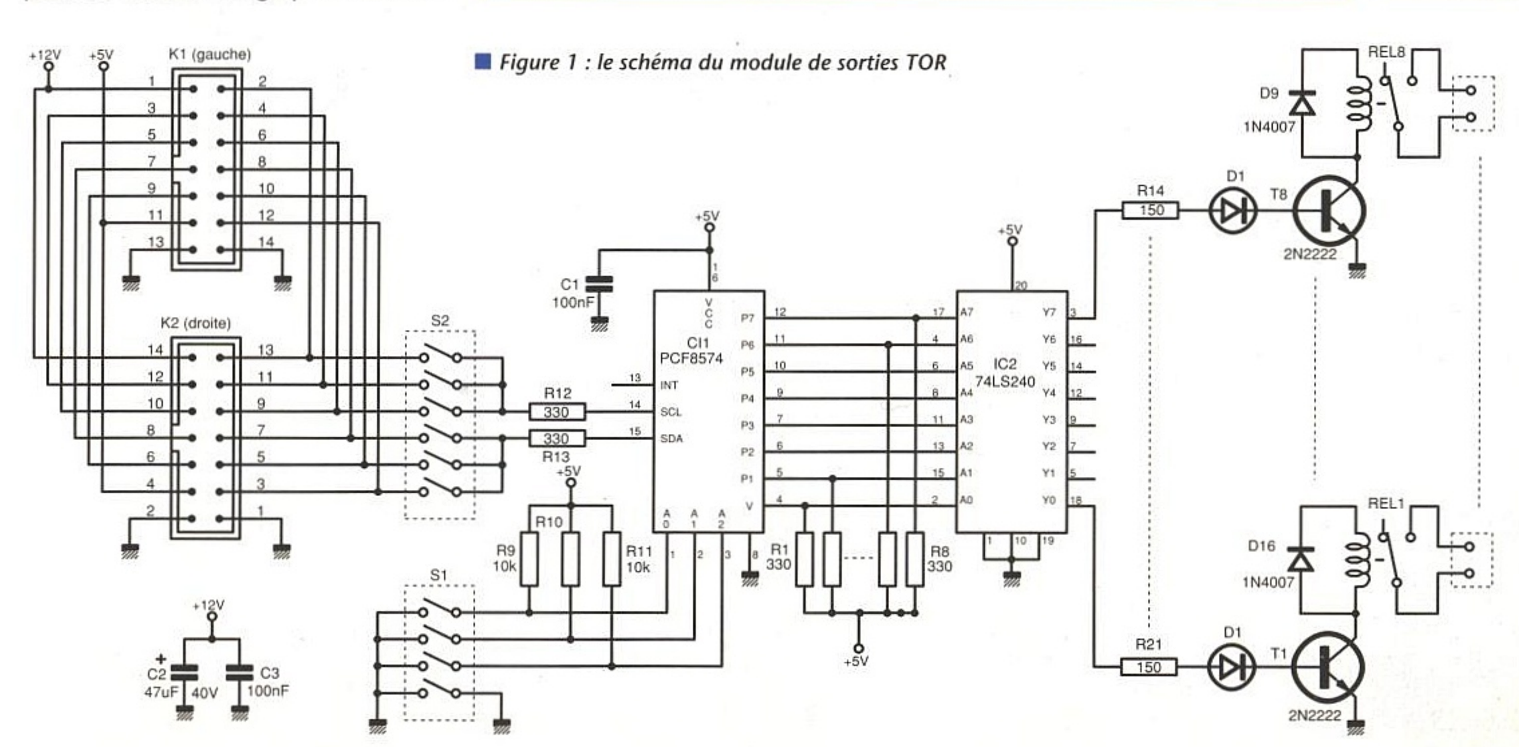


Schéma électrique

Le schéma électrique du bornier est donné **figure 4**. Il comprend deux alimentations distinctes (+5V et +12V), et une connectique de raccordement. Les alimentations n'appelent pas de commentaire particulier : transformateur, pont de diodes, filtrage par condensa-



■ *Figure 1 : le schéma du module de sorties TOR*



Guides-cartes mécaniques pour les borniers d'entrées-sorties

Les borniers d'entrées-sorties du système Domotique DOMESTICUS sont destinés à être fixés par des guides cartes au format Europe (100x160mm). Ces guides-cartes très pratiques peuvent être munis de flasques de deux types : des flasques à oeillets pour une fixation par vis, ou des flasques à montage sur rail électrique Oméga standard pour un montage dans une armoire électrique normalisée.

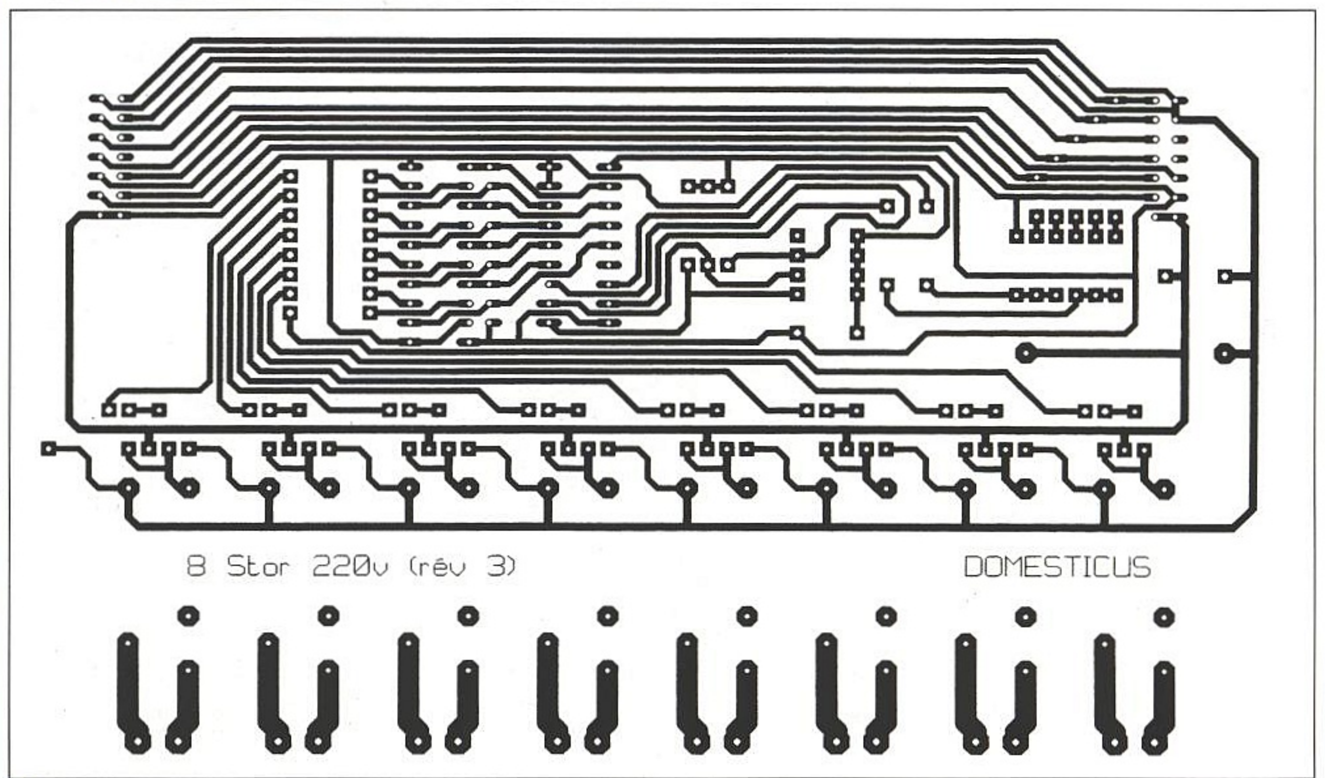


Figure 2a : CI module 8 sorties

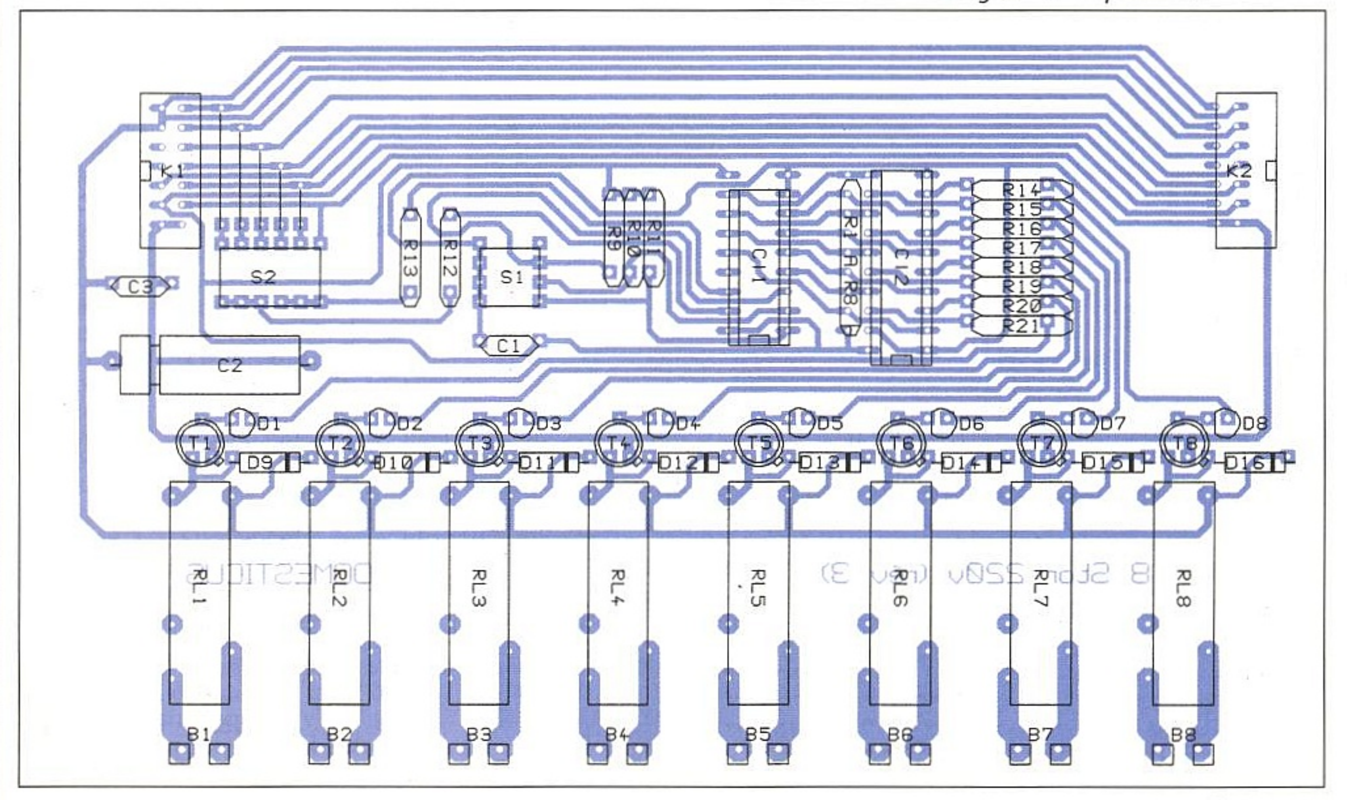
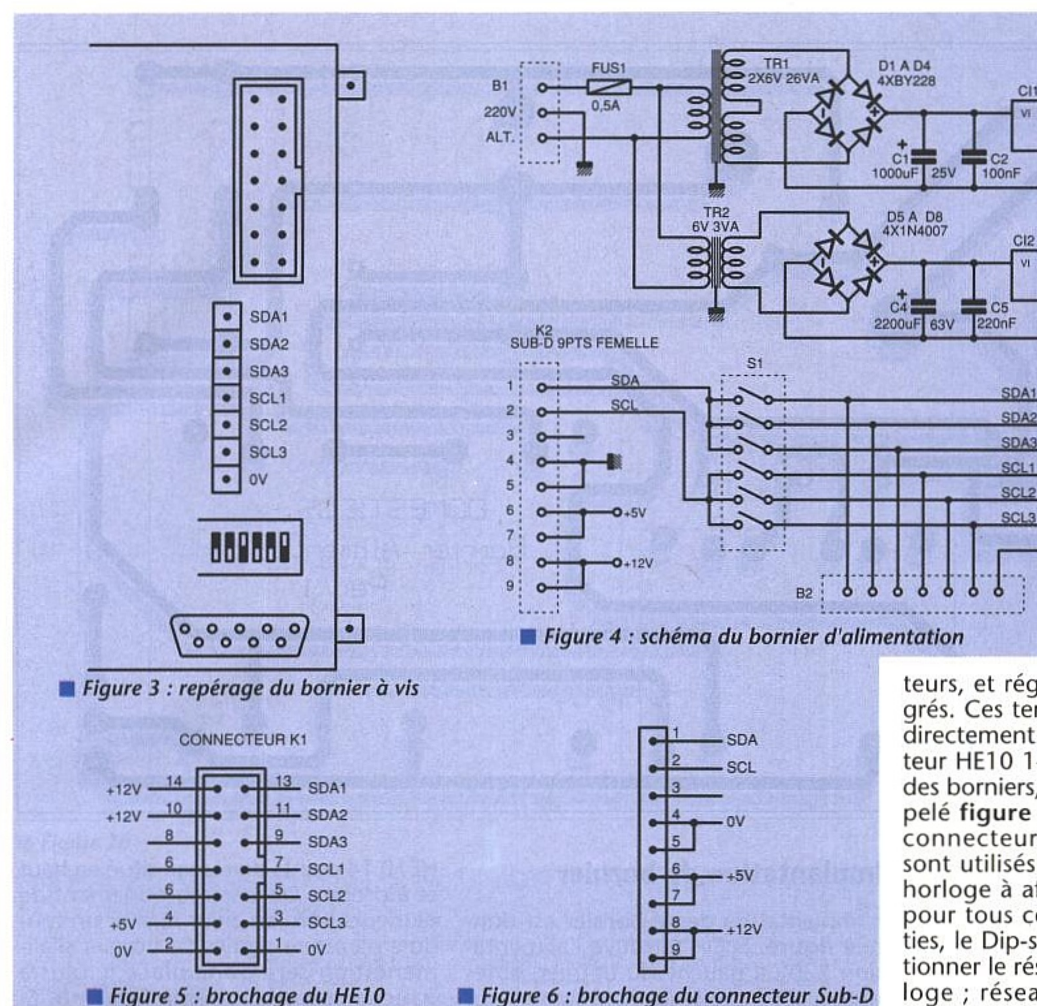


Figure 2b : l'implantation



■ *Figure 3 : repérage du bornier à vis*

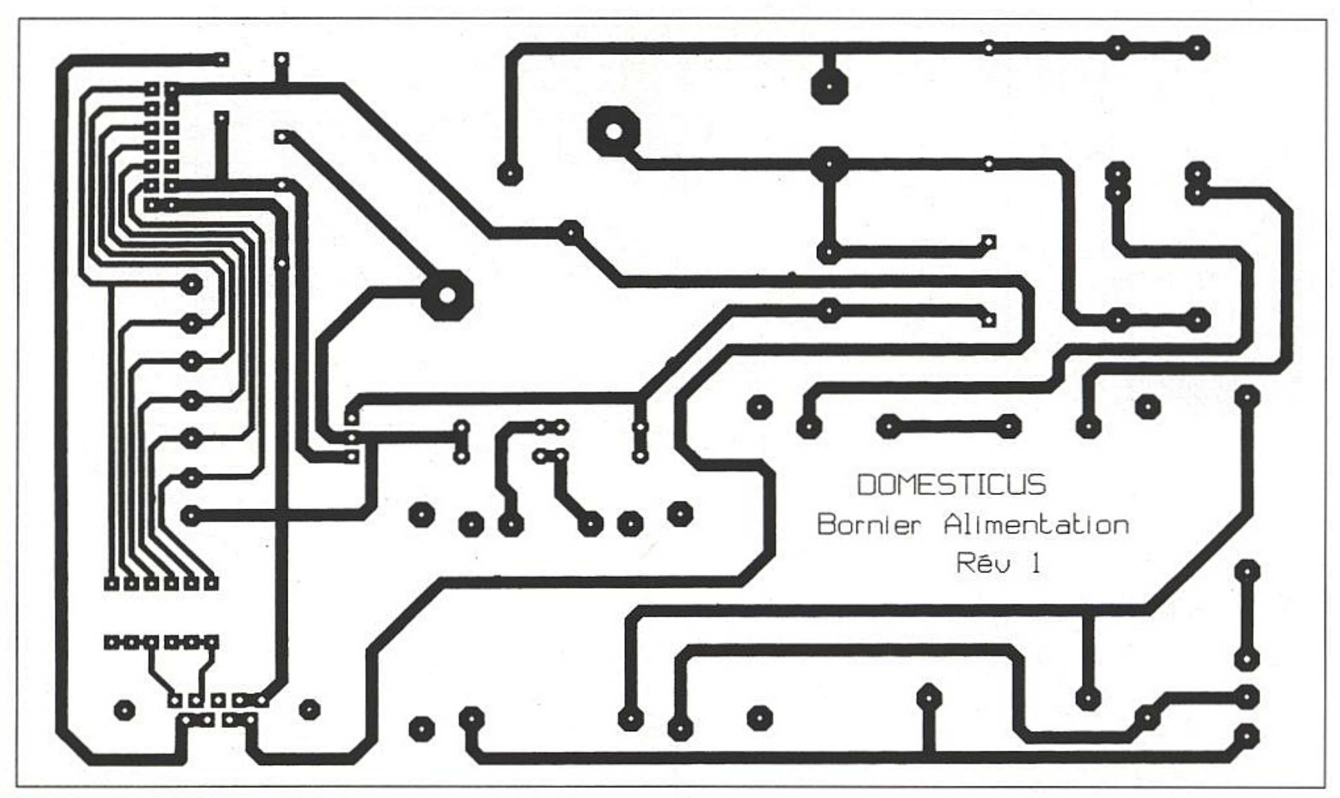
■ *Figure 4 : schéma du bornier d'alimentation*



■ *Figure 5 : brochage du HE10*

■ *Figure 6 : brochage du connecteur Sub-D*

teurs, et régulateurs de tensions intégrés. Ces tensions d'alimentation sont directement accessibles sur le connecteur HE10 14 points de raccordement des borniers, dont le brochage est rappelé **figure 5**. Le Dip-switch S1 et le connecteur Sub-D 9 points femelle sont utilisés pour connecter la future horloge à afficheurs géants. Comme pour tous ces borniers d'entrées-sorties, le Dip-switch S1 permet de sélectionner le réseau utilisé pour cette horloge ; réseau que l'on retrouve aux bornes 1 et 2 du connecteur Sub-D. La

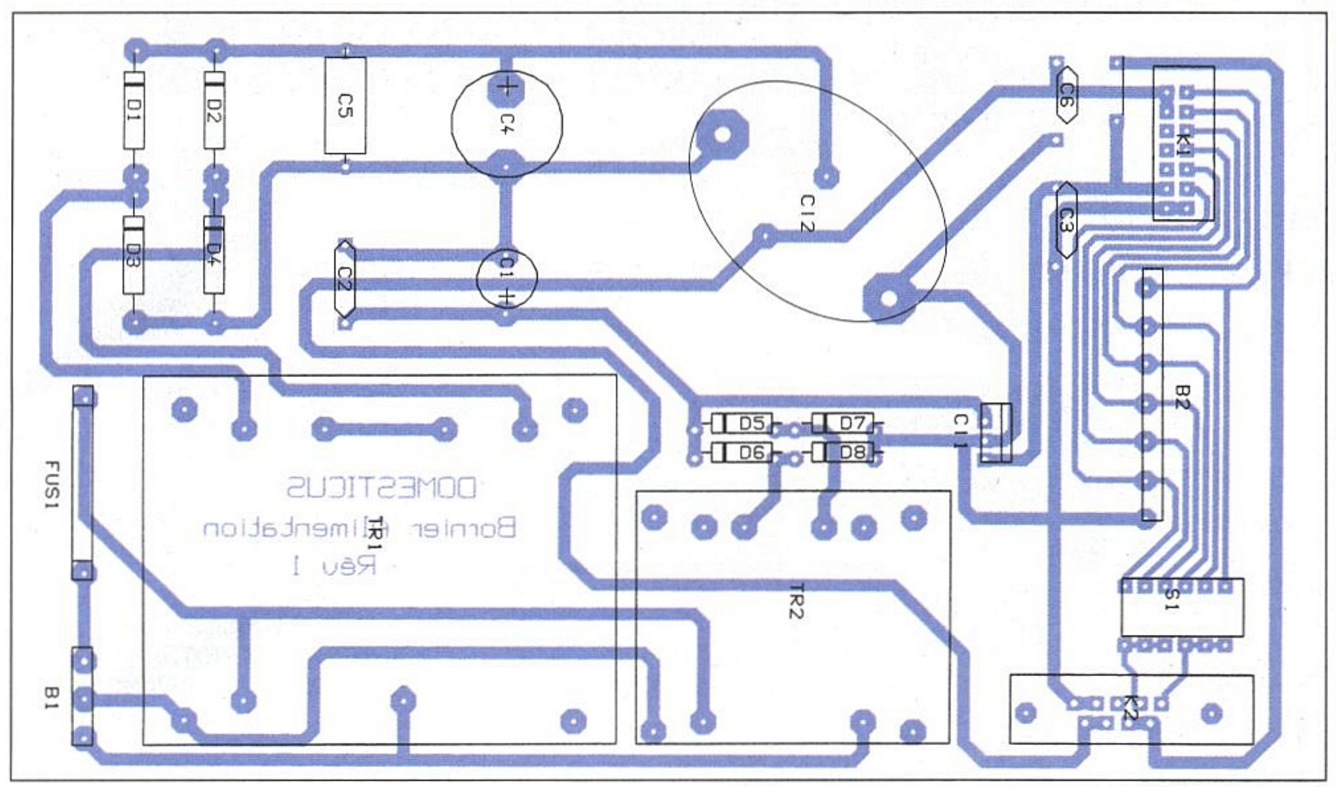


■ *Figure 7a*

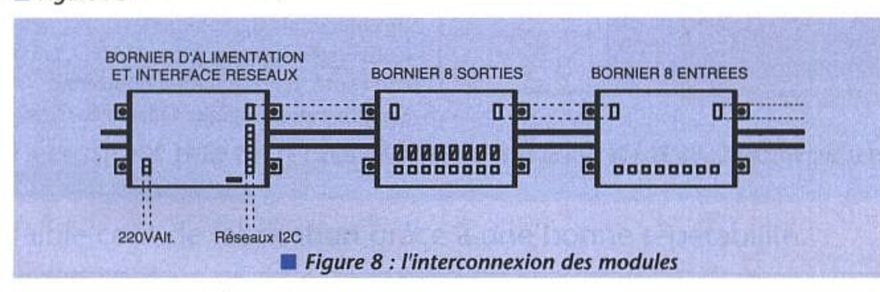
figure 6 indique le brochage de ce connecteur. Le raccordement 220V de ce bornier peut s'effectuer avec un cable à 3 fils (phase, neutre et Terre), pour permettre de relier les carcasses des transformateurs à la Terre.

Implantation du bornier

L'implantation de ce bornier est donnée **figure 7**. On retrouve l'alimentation 220V à gauche du bornier, et les réseau I2C à droite (avec le connecteur HE10 14 points du réseau situé en haut et à droite). Dans le regroupement de plusieurs borniers pour former un module d'entrées-sorties, le bornier d'alimentation sera donc placé le plus à gauche selon le schéma **figure 8**. Si



■ *Figure 7b*



■ *Figure 8 : l'interconnexion des modules*

vous n'avez pas prévu de connecter une horloge à afficheurs géants, vous n'avez pas besoin de cabler le Dip-switch S1, la Sub-D 9 points, et le strap à gauche du connecteur HE10 14 points.

Conclusion

Vous voici maintenant en possession des modules d'entrées-sorties digitales du système Domotique DOMESTICUS. Nous n'avons toujours pas abordé le problème du logiciel utilisé pour faire dialoguer ces borniers avec votre informatique. Ce sera chose faite dès le mois prochain, avec la publication de la carte d'interface PC, d'un exemple d'utilisation, et des routines de bas niveau pour les programmeurs.

J. GARBAY

NOMENCLATURE

Bornier 8 sorties Digitales

Résistances

R1 à R8 : Réseau SIL 8 x 330 Ω
R9,R10,R11 : 10 kΩ
R12,R13 : 330 Ω
R14 à R21 : 150 Ω

Condensateurs

C1,C3 : 100nF
C2 : 47µF 40 V

Semiconducteurs

D1 à D8 : LED 3mm rouges
D9 à D16 : 1N4007
T1 à T8 : 2N2222
IC1 : PCF8574
IC2 : 74LS240

Divers

Rel1 à Rel8 : Relais 12V - 1RT - 250V 10A pour montage sur circuit imprimé - Finder type 40, Matsushita, ou équivalent
K1,K2 : Connecteurs HE10 males 2 x 7 points bas profil sans verrou
S1 : Dipswitch horizontal 4 inters
S2 : Dipswitch horizontal 6 inters
B1 à B8 : Borniers doubles à vis au pas de 5,08 mm WEIDMULLER
1 support pour CI 16 broches
1 support pour CI 20 broches
1 profilé pour carte Europe 100 x 160 mm
2 plaques de fermeture pour montage direct ou montage sur rail Oméga

Bornier alimentation

Condensateurs

C1 : 1000 µF 25V radial
C2,C3,C6 : 100nF
C4 : 2200 µF 63V radial
C5 : 220nF

Semiconducteurs

D1 à D4 : BY228
D5 à D8 : 1N4007
IC1 : Régulateur intégré 7805 boitier TO220
IC2 : Régulateur intégré 7812 boitier TO3

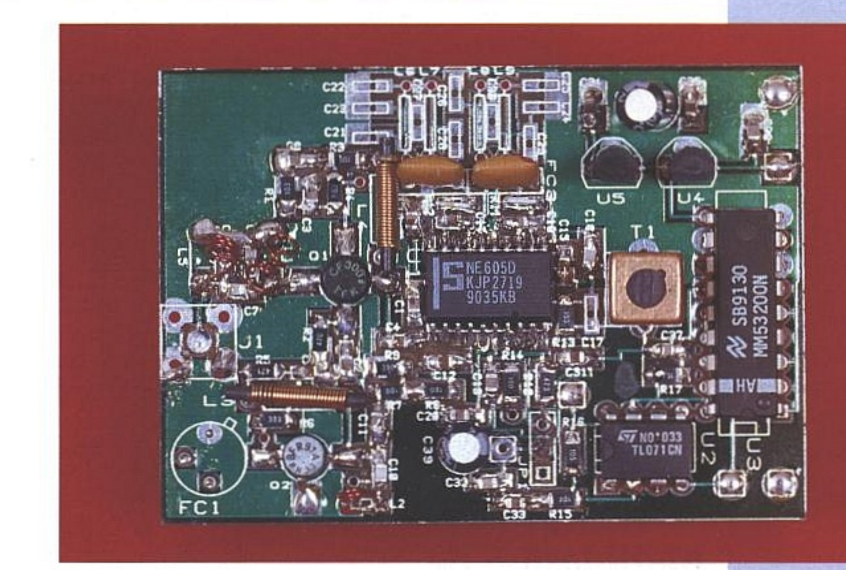
Divers

K1 : Connecteur HE10 mâle 2 x 7 points bas profil sans verrou
K2 : Sub-D 9 points femelle à souder
S1 : Dipswitch horizontal 6 inters
B1 : Bornier triple à vis au pas de 5,08 mm
B2 : Bornier sextuple à vis au pas de 5,08mm
TR1 : Transformateur 220 V - 2 x 6 V 26VA
TR2 : Transformateur 220 V - 1 x 6V 3VA
FU1 : Porte-fusible à souder sur CI
1 radiateur pour boitier TO3
1 fusible cartouche 5x20 500mA
1 profilé pour carte Europe 100 x 160 mm WEIDMULLER
2 plaques de fermeture pour montage direct ou montage sur rail Oméga



RÉCEPTEURS VHF À RÉSONATEUR À ONDES DE SURFACE

Dans un précédent numéro d'Electronique Radio-Plans, nous avons démontré que grâce aux résonateurs à ondes de surface Siemens, la conception d'émetteurs-récepteurs pouvait être simplifiée. Ces composants permettent la réalisation d'oscillateurs stables ne nécessitant pas de réglages. Cette caractéristique est évidemment un gage de faible coût de fabrication grâce à une bonne répétabilité.



Dans ce numéro, nous poursuivons notre enquête sur les résonateurs à ondes de surface Siemens. Nous atteindrons la gamme des 400 MHz et nous ferons un bref retour sur la bande 224,5 MHz. Nous traiterons les résonateurs cités dans le tableau de la **figure 1**.
Avec ces deux articles, les lecteurs disposent d'une schémathèque qui devrait permettre de répondre à la plupart des problèmes et des questions posées sur ce thème.
Le seul point faible de ces résonateurs est le calage en fréquence, et cette caractéristique doit être prise en compte au moment de la conception.
La légalisation d'une fréquence européenne à 433,92 MHz a donné naissance à certains résonateurs à ondes de surface destinés aux émetteurs-récepteurs travaillant dans cette bande.

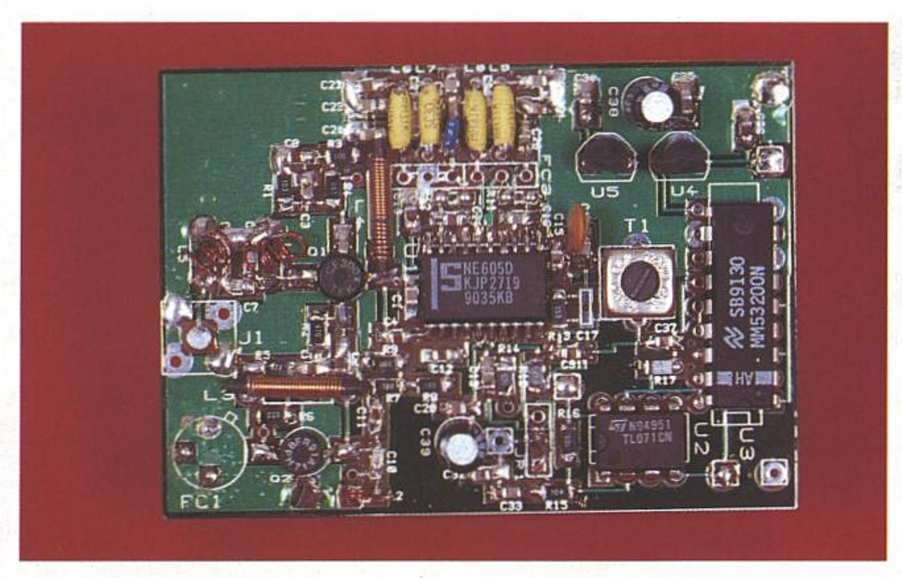
Différents types de résonateurs et circuits oscillateurs

Pour les résonateurs Siemens fonctionnant soit à 224 MHz soit à 433 MHz, on peut employer le schéma de principe de la **figure 2.** Ce schéma est intéressant car il utilise assez peu de composants et il offre une grande latitude quant au choix de la polarisation et donc de la puissance de sortie.
Les deux tableaux de la **figure 3** résument une partie des essais effectués avec cinq types de résonateurs et trois types de transistors.
Dans ces deux tableaux, on trouve la fréquence de l'oscillateur et la self L qui est associée puis les valeurs des résistances de polarisation et la puissance de sortie correspondante.
Le transistor BFT 24 est particulièrement intéressant pour son fonctionnement optimum avec un très faible courant, de l'ordre du mA. En contrepartie, la puissance délivrée à une charge de 50 Ω est faible et ne vaut que 10 µW.
Même si la puissance est faible, cette configuration reste intéressante pour des systèmes de transmission à courte ou très courte distance, et dans les systèmes où la consommation est un critère très important.
Pour des puissances supérieures, on a recours par exemple au BFR 34 qui permet de fournir 1 mW à une charge de 50 Ω avec le transistor polarisé avec un courant collecteur de 20 mA.
Finalement, avec un transistor comme le BFR 81 qui délivre soit 0,5 mW avec un courant de polarisation de 14 mA (V = 5 V), soit 5 mW en consommant 50 mA (V = 12 V), on couvre une plage de puissance répondant aux deux applications de transmission : émetteur et récepteur.
Le même étage oscillateur peut être employé dans l'émetteur et dans le récepteur : premier oscillateur local.
Pour l'oscillateur local, la puissance de sortie est choisie en fonction du type de mélangeur. Pour un circuit comme le NE 602 ou NE 612, on aura besoin d'environ 1 mW et pour un mélangeur à diodes type SBL 1X mini-circuits on prendra 5 mW, + 7 dBm.
La structure de l'oscillateur de la **figure 2** pourra être appliquée à d'autres types de transistors. On prendra garde à choisir une fréquence de transition ft compatible avec la fréquence d'oscillation.
Si la simplicité, versatilité et reproductibilité de cet oscillateur sont intéressantes, il n'en est pas de même de la pureté spectrale.
En effet, en général, l'harmonique 2 est

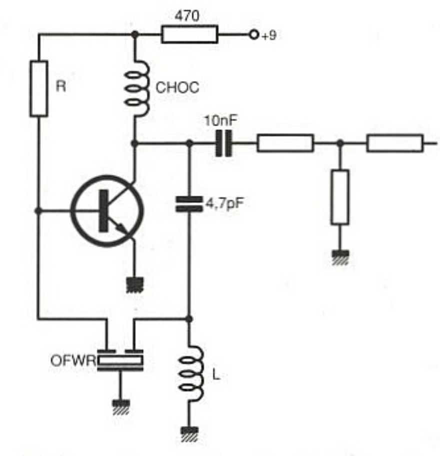
Figure 1

Référence Siemens	f en MHz	Tolérance kHz
OFW R 2531	423,22	± 75
OFW R 2632	433,92	± 100
OFW R 2633	434,32	± 75
OFW R 2523	224,50	± 67
OFW R 2637	213,80	± 64





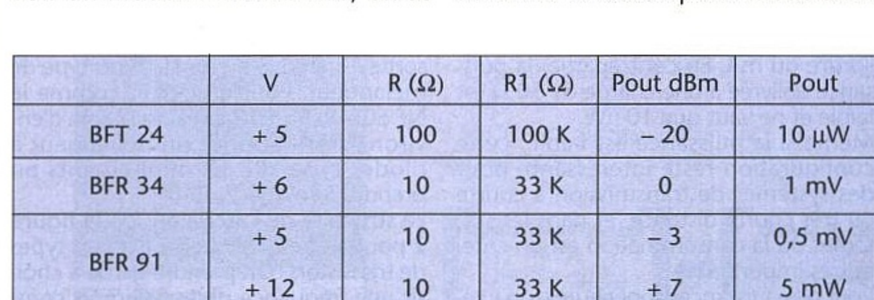
au même niveau que le fondamental et l'harmonique 3 est environ à – 20 dB.
Dans un but d'optimisation, on pourrait chercher le transistor ayant le ft juste suffisant pour permettre l'oscillation et rejeter les harmoniques au plus bas niveau. Cette optimisation s'effectue non seulement par le choix du composant mais aussi par le choix du point de polarisation.



■ *Figure 2 : oscillateur simple à résonateur*

Modulation de l'émetteur

L'oscillateur de la **figure 2** ne peut pas être modulé en fréquence, cette interdiction découle du très fort coefficient de surtention du résonateur qui lui donne une grande stabilité.
Une modulation d'amplitude peut être envisagée soit sur la base, soit sur le collecteur, le tout consistant bien sûr à modifier le courant de polarisation et donc la puissance de sortie.
Dans le cas de la transmission de données, il est inutile de chercher à concevoir un modulateur linéaire, on se contente donc d'une structure tout ou rien telle celle de la **figure 4**.
L'entrée modulation peut recevoir le signal de sortie d'un circuit de codage comme le MM 53200.
Il est connu qu'une transmission en modulation de fréquence donne de meilleurs résultats que la modulation d'amplitude. Si la qualité de la transmission est le critère le plus important, on opte pour la FM.
Dans ces conditions, le résonateur à ondes de surface Siemens ne peut plus être utilisé. On peut employer un VCO Murata qui facilite la conception. Pour diminuer le coût, il est aussi possible de concevoir un VCO à partir de résonateurs diélectrique Siemens en λ /4.
Nous reviendrons probablement sur ce sujet dans un autre article.
Dans ce numéro, nous nous intéresserons plus particulièrement aux récepteurs 433,92 MHz fonctionnant en modulation d'amplitude. Les mesures effectuées en modulation de fréquence ne sont données qu'à titre de comparaison.
L'établissement du schéma de l'émetteur, à partir des données de la **figure 4** ne pose pas de problème particulier. Toutefois, à titre d'exemples, nous donnerons une application à 224,5 MHz.



	V	R (Ω)	R1 (Ω)	Pout dBm	Pout
BFT 24	+ 5	100	100 K	– 20	10 µW
BFR 34	+ 6	10	33 K	0	1 mV
BFR 91	+ 5	10	33 K	– 3	0,5 mV
	+ 12	10	33 K	+ 7	5 mW

■ *Figure 3a : Puissance de sortie en fonction du transistor et de la polarisation.*

Type de résonateur Siemens	F en MHz	L(nH)
OFW 2531	423,22	15
OFW 2632	433,92	15
OFW 2633	434,32	15
OFW 2523	224,5	40
OFW 2637	213,8	40

■ *Figure 3b : Self d'adaptation en fonction de la fréquence.*

Les différents types de récepteurs

Nous avons sélectionné deux fréquences : 224,5 MHz et 433,92 MHz. Ces deux fréquences sont des fréquences porteuses **légales** et **autorisées.**
Les trois autres fréquences contenues dans le tableau de la figure 1 sont uniquement destinées aux récepteurs.
Le résonateur R 2637 entre dans la réalisation de l'oscillateur local à 213,8 MHz soit 224,5 MHz - 10,7 MHz.
La fréquence intermédiaire vaut 10,7 MHz et le calage des résonateurs étant à ± 70 MHz environ, la largeur de bande du filtre FI sera supérieure à 140 kHz.
Les résonateurs R 2531 et R 2633 sont destinés aux oscillateurs locaux pour la réception de la fréquence à 433,92 MHz.
Dans le premier cas, la FI vaut 10,7 MHz et la largeur de bande FI sera supérieure à 175 kHz.
Dans le second cas, la FI vaut 400 kHz.

Structure du récepteur

Dans un premier temps, nous effectuerons des tests sur deux récepteurs conformes au schéma synoptique de la **figure 5.**
Il s'agit de deux récepteurs à un seul changement de fréquence.
La fréquence intermédiaire prend deux valeurs, soit 10,7 MHz, soit 400 kHz.
Le schéma synoptique du récepteur est très classique : étage d'entrée à faible bruit, mélangeur, amplification FI et démodulation.
Le mélange, l'amplification Fi et la démodulation sont confiés au circuit intégré NE 605. La sortie RSSI est employée en tant que sortie AM.
Le démodulateur FM n'est mis en place que dans un but de mesure et n'est pas utilisée dans l'application initiale.
A l'entrée un filtre LC sélectionne la fréquence à recevoir et élimine la fréquence image. Si l'on souhaite une meilleure rejection de la fréquence image, on peut ajouter un filtre hélicoïdal.



■ *Figure 6 : schéma générique de récepteurs à ROS.*

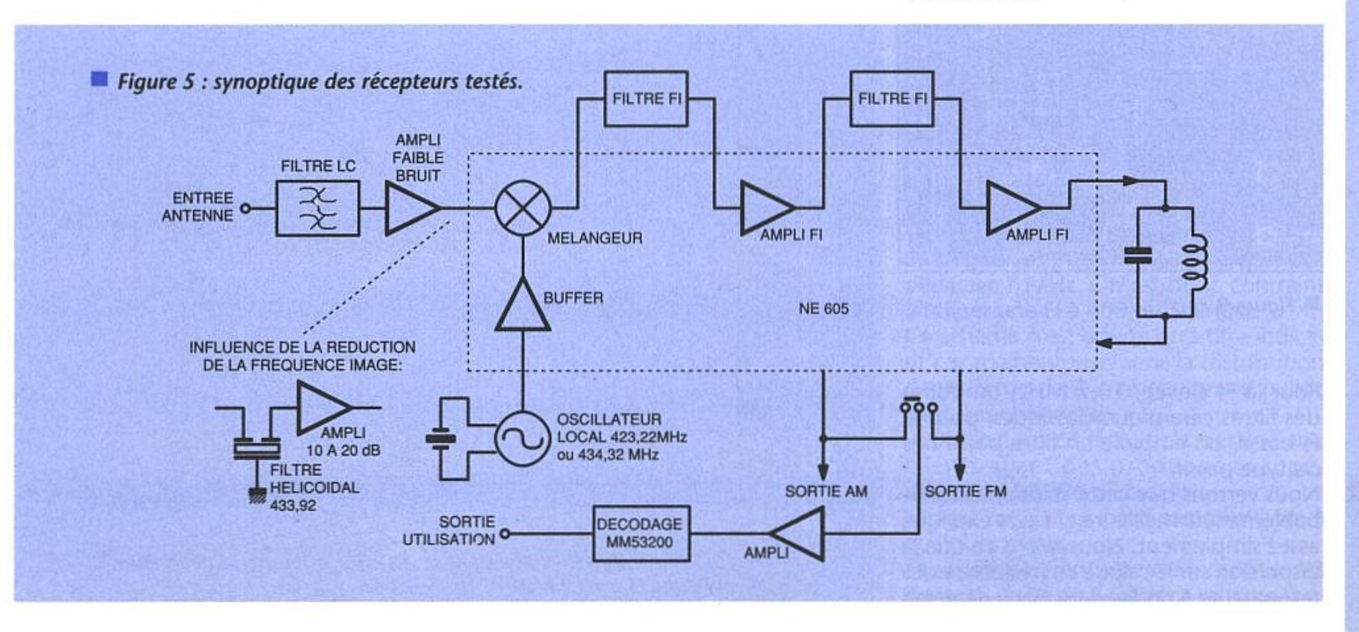
Figure 4 : oscillateur module en amplitude tout ou rien.

Schéma du récepteur

Le schéma du récepteur est donné à la **figure 6.** Le filtre d'entrée est un filtre en PI. Le signal est amplifié par un étage à faible bruit : CF 300. Le signal est ensuite injecté à la broche 1 du NE 605.

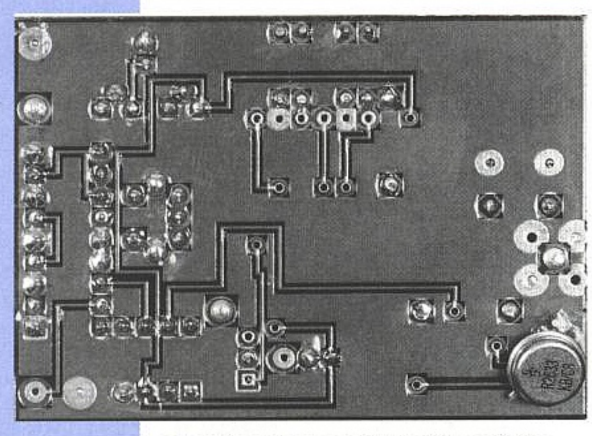
L'oscillateur local est bâti autour du transistor T2. Le signal de sortie de cet oscillateur est envoyé à la broche 4 du NE 605.

Si l'oscillateur local est à la fréqunce du 423,22 MHz, la fréquence intermédiaire vaut 10,7 MHz et si cette fréquence est à 434,92 MHz, la fréquence intermédiaire vaut 400 kHz.



■ *Figure 5 : synoptique des récepteurs testés.*



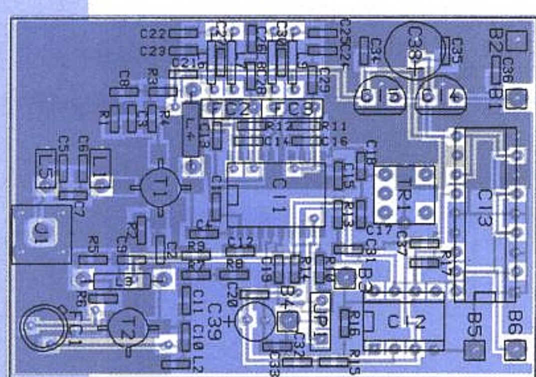


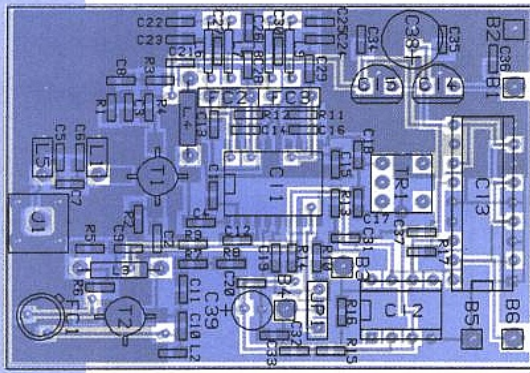
■ *Le résonateur est placé côté soudures.*

■ *Figure 7*



■ *Figure 8*





■ *Figure 9*

Pour la FI élevée, 10,7 MHz, on utilise des filtres céramique classiques, pour la FI basse, on emploie un filtre passe-bas du type passif.

Nous verrons que cette solution est probablement la meilleure et ceci s'explique assez simplement. Nous avons vu que la dispersion sur le calage en fréquence des résonateurs à ondes de surface détermi-nait la tolérance sur la FI : 90 kHz environ.

Sur le schéma de la figure 6, les filtres passifs mis en service dans le cas de la FI à 400 kHz sont encadrés par un pointillé.

Dans les deux cas, le signal résultant d'une pseudo-démodulation AM apparaît à la broche 7 et le signal résultant de la démodulation FM à la broche 9.

Un cavalier JP1 permet de choisir l'un ou l'autre mode.

Le signal est filtré, amplifié et envoyé au circuit MM 53200.

Le circuit MM 53200 est placé ici à titre d'exemple, on pourrait envisager bien d'autres solutions.

RÉALISATION PRATIQUE

Un seul circuit imprimé réunit les deux configurations : FI à 10,7 MHz ou FI à 400 kHz.

Le tracé des pistes côté soudures est donné à la **figure 7,** côté composants à la **figure 8** et l'implantation correspondante est donnée à la **figure 9.**

Nous avons réalisé une maquette pour chaque type de récepteur et nous nous proposons maintenant d'examiner les résultats des premières mesures.

On suppose, bien entendu, que les circuits ont été câblés convenablement. En principe le câblage ne pose aucun problème si l'on procède avec soin.

Bien entendu TR1 et C17 sont fonction de la fréquence intermédiaire et ce circuit oscillant résonne à ladite fréquence.

FI haute : 10,7 MHz

En modulation de fréquence, le signal modulant vaut 1 kHz et l'excursion en fréquence 50 kHz. En modulation d'amplitude le signal modulant vaut 1 kHz et l'indice de modulation 60 %.

Le résultat des mesures du rapport signal sur bruit en fonction de la valeur du signal d'entrée est condensé dans le tableau de la **figure 10** et les courbes de la **figure 11** reprennent ces résultats.

Pin dBm	S/B AM dB	S/B FM dB
– 50	48	61
– 60	47	60
– 70	43	*
– 80	34	47,0
– 85	30,6	42,1
– 90	25,5	38
– 95	19,8	33,7
– 100	14,3	28,4
– 102	12,3	*
– 104	10,3	18,0
– 106	8,2	15,3
– 107	7,3	13,0
– 110	4,5	6,6

Figure 10 : filtre FI de type céramique FI = 10,7 MHz avec f = 1 KHz m = 60 % en AM, Δf = 10 kHz en FM

Ces deux courbes mettent en évidence la supériorité de la modulation de fréquence sur la modulation d'amplitude lorsque l'on a un indice de modulation FM Δf/f suffisamment élevé.

En modulation d'amplitude et en transmission de données, on peut avoir de très bons résultats avec seulement 2 µV à l'entrée du récepteur : 2 µV (– 104 dBm), S/B = 10 dB environ.

Pour augmenter les performances du circuit on peut toujours envisager l'adjonction d'un étage amplificateur supplémentaire entre le CF 300 et le circuit NE 605.

FI basse : 400 kHz

Pour la fréquence intermédiaire basse : 400 kHz, les caractéristiques de la modulation d'amplitude ne changent pas : fréquence de modulation 1 kHz et indice de modulation 60 %.

Par contre en FM, le coefficient de sur-

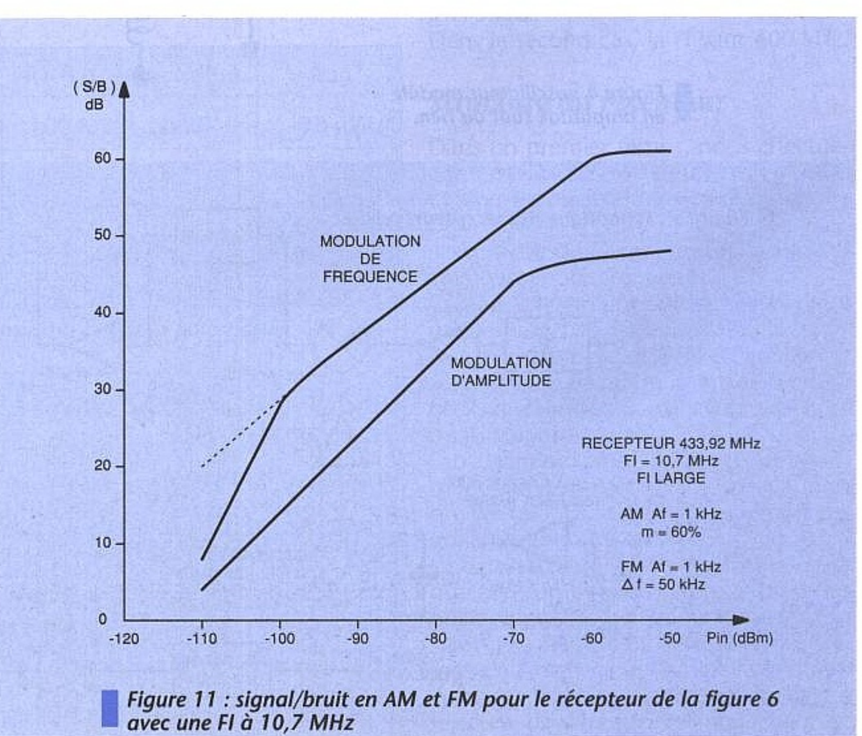
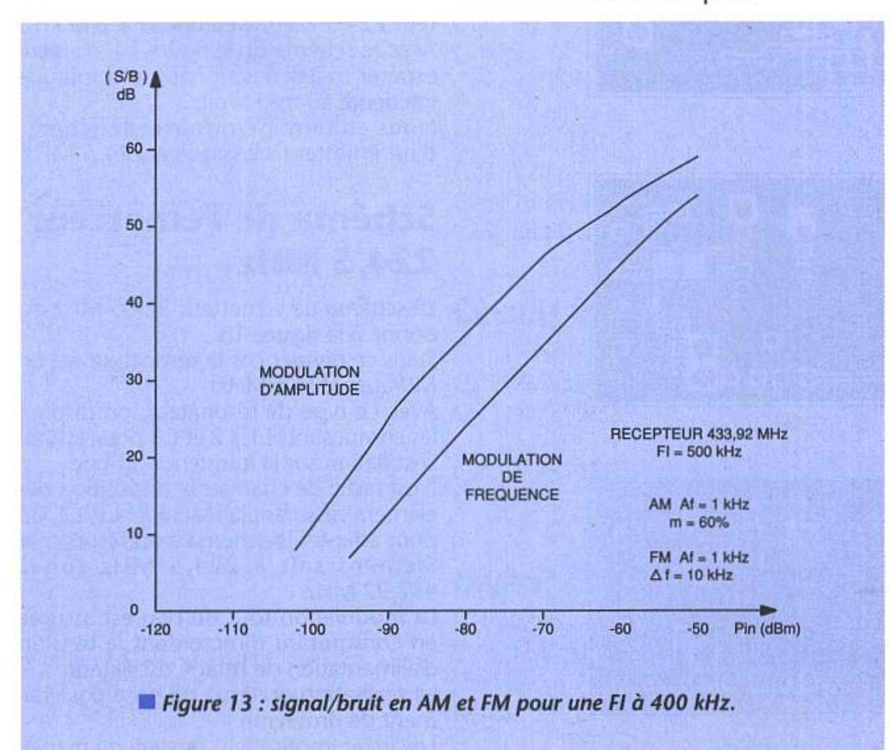


Figure 11 : signal/bruit en AM et FM pour le récepteur de la figure 6 avec une FI à 10,7 MHz

■ *Figure 14 : récepteur à UAA 3201T.*

Pin dBm	S/B AM dB	S/B FM dB
– 50	59	54
– 60	53	45
– 70	44	37
– 80	35	24
– 85	31	18,3
– 90	22	12,8
– 95	17,5	7,0
– 100	11	4,8
– 102	8,3	*

Figure 12 : filtre FI de type passif avec FI = 400 kHz
f = 1 KHz
m = 60 % en AM,
Δf = 10 kHz en FM



■ ***Figure 13 : signal/bruit en AM et FM pour une FI à 400 kHz.***

tension du circuit déphaseur TR1, C17 nous contraint à prendre un indice de modulation plus faible : Δf = 10 kHz f = 1 kHz.

Dans ce cas, la largeur de bande du filtre FI n'est pas adaptée à la modulation. Les résultats sont donnés dans le tableau de la **figure 12** et de la **figure 13** traduit ces chiffres en deux courbes.

Dans ce nouveau cas, la modulation d'amplitude s'avère meilleure que la modulation de fréquence.

En AM pour un rapport signal sur bruit de 10 dB le niveau d'entrée doit être de l'ordre de 4 μV.

De meilleurs résultats seraient obtenus en optimisant la largeur du filtre FI, en tenant compte simultanément de la dispersion du calage et de la largeur de bande réellement occupée.

Des essais pourraient être tentés en remplaçant les filtres FI passifs par des transformateurs moyenne fréquence 455 kHz décalés à 400 kHz et fortement amortis par une résistance externe.

Le principal désavantage de cette dernière solution est la présence de composants ajustables.

A ce stade de la réflexion, on peut remarquer que les résonateurs à ondes de surface sont donc destinés aux très grandes séries, là où le coût est plus important que les performances.

On préfère donc se contenter de performances moyennes dues à la dispersion sur le calage et à la largeur des filtres FI qui en résulte. En contrepartie aucun réglage en fin de chaîne n'est nécessaire et le coût est un conséquence.

Nous passons maintenant au troisième récepteur que nous avons expérimenté.

Récepteur à UAA 3201T

Le schéma de principe du récepteur bâti autour du circuit Philips UAA 3201T est représenté à la **figure 14.**

Le circuit UAA 3201T est un circuit récepteur pour des fréquences d'entrée comprises entre 150 et 450 MHz. Ce circuit est optimisé pour une basse consommation : 3,7 mA en moyenne sans 5 V et 5,1 mA maxi.

La fréquence intermédiaire ne doit pas dépasser 1 MHz. On est donc contraint d'utiliser une FI à 400 ou 500 kHz.

Ce schéma n'est donné qu'à titre indicatif car malheureusement la distribution de ce circuit dans le réseau grand public n'est pas envisagée.

Pour l'instant ce circuit est destiné aux clients ayant des productions supérieures à 100 000 pièces.

Malgré tout, nous nous sommes livrés à quelques essais.

Le schéma de principe du récepteur est donné à la **figure 14.** Sur ce schéma l'ac-

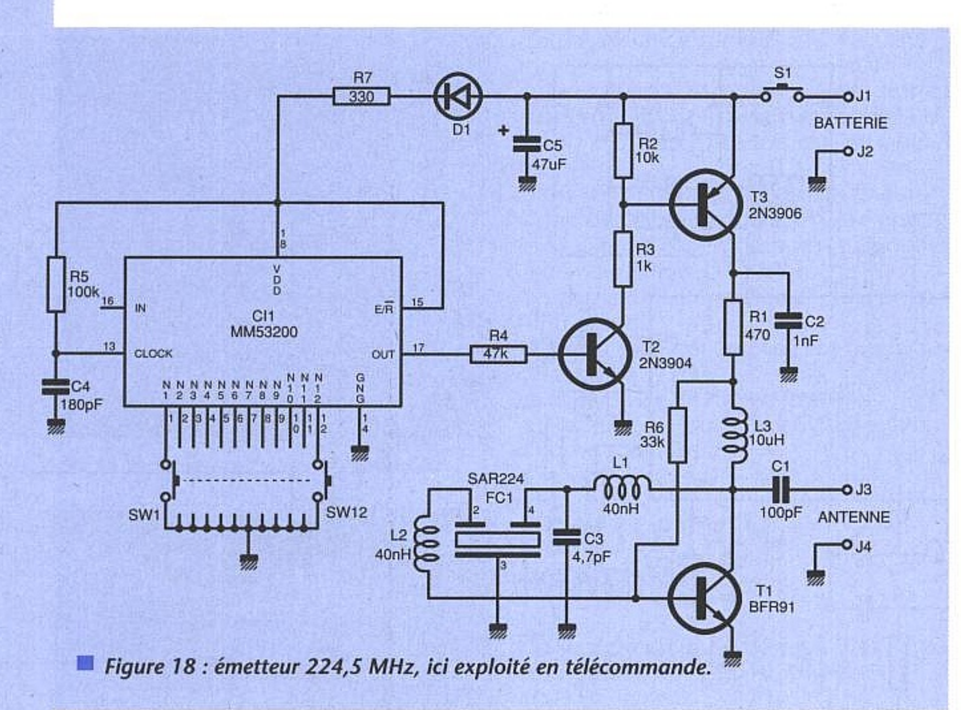
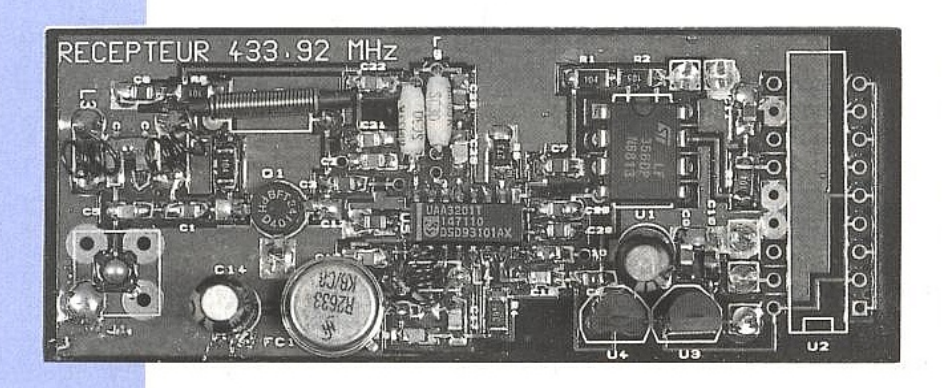
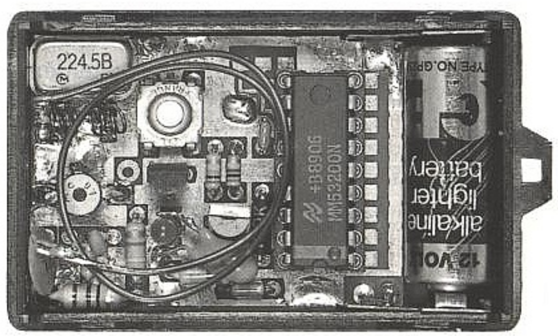


Figure 18 : émetteur 224,5 MHz, ici exploité en télécommande.

cent a été mis sur la consommation.
L'étage amplificateur d'entrée est à faible consommation : BFT 24.
L'amplificateur opérationnel LF 356 peut être remplacé par un modèle à faible consommation. Dans certains cas on peut envisager de se passer de l'amplificateur.
Un prototype a été réalisé à partir du tracé des pistes côté soudures donné à la **figure 15**, côté composants à la **figure 16** et l'implantation des composants à la **figure 17.**
L'emploi de composants CMS même s'il n'est pas généralisé permet la réalisation d'un récepteur de très faibles dimensions.
Sur le schéma de la figure 14, l'oscillateur est du type base commune. Les deux points chauds du résonateur à ondes de surface sont bien réunis ensemble, il ne s'agit pas d'une erreur de dessin.
La faible valeur de la fréquence intermédiaire ne permet pas une rejection de la fréquence image.
Le circuit UAA 3201 T fonctionne de 150 à 450 MHz.





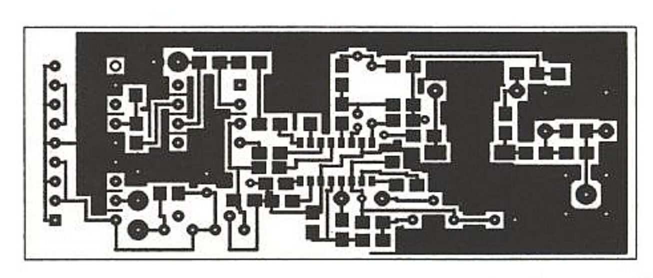


Figure 15

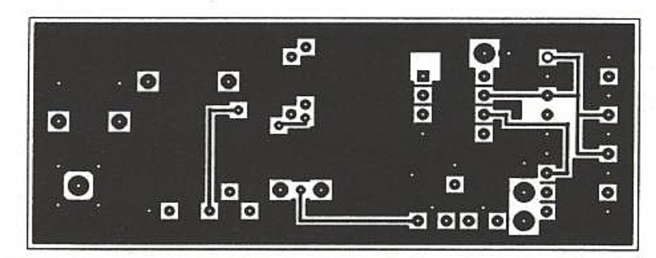


Figure 16

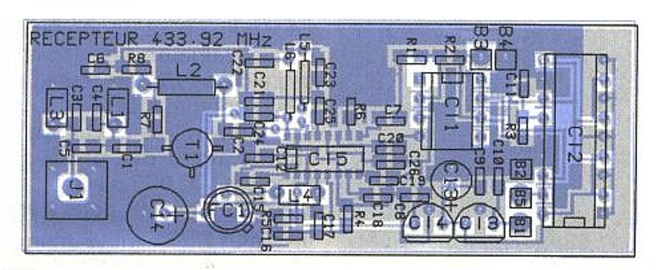


Figure 17

On peut facilement transposer le schéma de la figure 14 à 224,5 MHz.
Il n'existe pas de résonateur à 224,1 ou 224,9 MHz, mais il existe un résonateur à 224,7 MHz, RFM.
On pourrait donc envisager un récepteur 224,5 MHz avec une FI à 200 kHz.
Avec le schéma de la figure 14, on peut espérer traiter des signaux d'amplitude inférieure au microvolt.
Nous en terminerons avec le schéma d'un émetteur classique à 224,5 MHz.

Schéma de l'émetteur 224,5 MHz

Le schéma de l'émetteur 224,5 MHz est donné à la **figure 18.**
Dans ce dernier cas le résonateur est un Murata à 224,5 MHz.
Avec ce type de résonateur, on dispose les composants L1, L2 et C3 pour assurer l'oscillation sur la fréquence ad-hoc.
Il est facile de changer la disposition des éléments assurant la réaction : L1, L2, C3 pour adapter le schéma à un résonateur Siemens soit à 224,5 MHz soit à 433,92 MHz.
La modulation tout ou rien est assurée en commutant directement la tension d'alimentation de l'étage oscillateur.
La réalisation pratique ne pose pas vraiment de problème.
Le circuit imprimé est dessiné de maniè-



re à s'insérer parfaitement dans un boîtier destiné à cette application.
La tracé des pistes côté soudures est donné à la **figure 19**, côté composants à la **figure 20** et l'implantation correspondante à la **figure 21**.
Le boîtier spécifique ainsi que le résonateur Murata sont disponibles notamment chez Selectronic.
Il n'y a pas de récepteur spécifique prévu pour cet émetteur. Un des trois modèles présenté précédemment devra être adapté à 224,5 MHz. Ceci ne pose vraiment pas de problème puisqu'il s'agit de changer un résonateur céramique, la self dans le circuit de réaction et le filtre en PI dans l'entrée du récepteur.

Conclusion

Dans ces quelques pages, nous avons démontré que les résonateurs à ondes de surface permettaient de réaliser très simplement des émetteurs-récepteurs miniatures si l'on acceptait de faire quelques concessions quant aux performances.
Ces composants sont donc voués à prendre une place de plus en plus importante pour les télécommandes d'ouverture de nos portes de voiture et autres systèmes de téléalarme, télémesure ou domotique.

F. de DIEULEVEULT

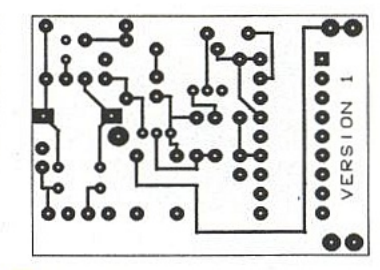
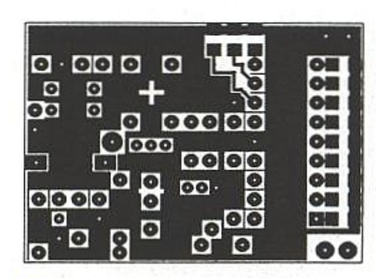


Figure 19


Figure 20

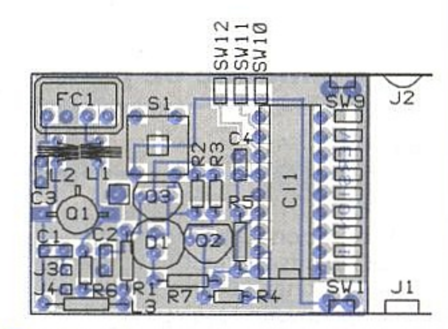


Figure 21

NOMENCLATURE
(figure 6)

Résistances
R1 : 10 kΩ
R2 : 47 Ω
R3, R14 : 10kΩ
R4 : 6,8 kΩ
R5, R11, R12 : 470 Ω
R6 : 33 kΩ
R7, R8 : 18 Ω
R9 : 39 Ω
R10 : 47 kΩ
R13 : 15 kΩ
R17, R15 : 100 kΩ
R16 : 1 MΩ

Condensateurs
C1, C2, C3, C8, C9, C11, C12, C37 : 1 nF
C4 : 1,5 nF
C5, C6, C18 = 1,5 pF
C7 : 2,2 pF
C10 : 4,7 pF
C13, C14, C15, C16, C19, C20, C28, C29, C31, C32, C33, C34, C35, C36 : 100 nF
C17 : 680 pF
C21, C23, C24, C26 : 150 pF
C25, C22 : 220 pF
C27, C30 : 15 pF
C38, C39 : 10 µF

Semiconducteurs
T1 : CF300
T2 : BFR91

Circuits intégrés
CI1 : NE 605
CI2 : LF356 ou TL071
CI3 : MM53200
CI4 : LM7809KC
CI5 : LM7805KC

Divers
FC1 : R2637
FC2, FC3 : 10,7 MHz (E 10,7A)
L1, L5 : 40 nH
L2 : 39 nH
L3 : 10 µH
L4 : choc
L6, 67, L8, L9 : 330 µH
TR1 : LMC 4102 (400 kHz)
ou 2SK241 (10,7 MHz)

NOMENCLATURE
(figure 14)

Résistances
R1, R3, R7 : 100 kΩ
R2 : 1 MΩ
R4 : 390 kΩ
R5 : 270 Ω
R6 : 27 kΩ
R8 : 100 Ω

Condensateurs
C1, C6, C11 : 1 nF
C2 : 1,5 nF
C3, C4 = 1,5 pF
C5 : 2,2 pF
C7, C8, C9, C10, C26 : 100 nF
C13, C14 : 10 µF
C15 : 56 pF
C16 : 3,9 pF
C17 : 4,7 pF
C18 : 3,3 nF
C19 : 2,2 nF
C20 : 10 nF
C24, C21 : 150 pF
C22 : 220 pF
C23 : 18 pF
C25 : 68 pF

Circuits intégrés
CI1 : LF356
CI2 : MM53200
CI3 : LM7809KC
CI4 : LM7805KC
CI5 : UAA3201T
T1 : BFT24

Divers
FC1 : R2633
L1, L3, L4 : 40 nH
L2 : choc
L6, L5 : 330 µF

NOMENCLATURE
(figure 18)

Résistances
R1 : 470 Ω
R2 : 10 kΩ
R3 : 1kΩ
R4 : 47 kΩ
R5 : 100 kΩ
R6 : 33 kΩ
R7 : 330 Ω

Condensateurs
C1 : 100 µF
C2 : 1 nF
C3 : 4,7 pF
C4 : 180 pF
C5 : 4,7 µF

Semiconducteurs
T1 : BFR91
T2 : 2N3904
T3 : 2N3906
CI1 : MM53200

Divers
FC1 : SAR224
L2, L1 : 40 nH
L3 : 10 nH
S1 : poussoir miniature

L'OUTIL DE DÉVELOPPEMENT PICSTART 16 B

Un outil de développement, même très simple, c'est-à-dire réduit à un assembleur, est un minimum nécessaire pour qui veut travailler avec des microcontrôleurs, même de façon occasionnelle. En effet, s'il est concevable d'assembler "à la main" lorsque l'on en est au tout début de son initiation à la micro-informatique, une telle pratique est inutilisable dès que l'on veut écrire efficacement des programmes qui dépassent quelques dizaines de lignes. Si cet outil comporte en plus un simulateur, les conditions de travail deviennent très confortables puisqu'il est alors possible d'essayer la majeure partie de ses programmes sans câbler la moindre maquette. C'est un tel ensemble, et même un peu plus, que nous propose Microchip avec son PICSTART 16 B commercialisé pour 1500 Francs environ au moment où ces lignes sont écrites.



Le contenu du PICSTART 16B

Malgré son prix relativement bas pour un produit de ce type, le contenu du PICSTART 16 B est apte à satisfaire l'utilisateur exigeant. Il contient en effet :
- Un assembleur pour les microcontrôleurs de la série PIC 16CXX,
- Un simulateur pour ces mêmes circuits,
- Un logiciel de commande de programmateur,
- Une petite carte de programmation à connecter sur le port série de votre micro-ordinateur,
- Et un échantillon de PIC 16C55 et de PIC 16C71 en boîtier céramique à fenêtre (donc effaçable) s'il vous plait !

Tous ces logiciels sont prévus pour fonctionner sur n'importe quel compatible PC et ne sont pas très exigeants puisqu'ils se satisfont de 640 K de RAM et d'un DOS supérieur ou égal au 3.3.

Deux gros data-books complètent cet ensemble avec celui consacré aux fiches techniques des circuits Microchip et celui contenant de nombreuses notes d'application fort intéressantes et bien documentées. Une notice, très mince par contre, sert de mode d'emploi au PICSTART 16 B.

On y découvre en fait le mode d'installation du logiciel qui se résume à une copie des fichiers (non protégés) dans le répertoire du disque dur de son choix ainsi que le mode d'emploi du programmateur. Pour ce qui est de l'assembleur et du simulateur, il vous faudra faire chauffer votre imprimante car ils sont contenus chacun dans un fichier texte se trouvant sur la disquette. Cette solution, discutable pour certains, permet toutefois de réduire les coûts de production du produit en économisant les frais d'impression de



deux manuels d'une centaine de pages chacun. Pour un ensemble vendu 1500 Francs, c'est une solution acceptable.

Ce qui manque dans l'emballage....

Pas grand chose certes mais il vaut mieux le savoir si vous faites vos emplettes par correspondance; cela vous évitera de devoir passer une deuxième commande après avoir reçu votre PICSTART !

Il faut prévoir une alimentation délivrant 9 volts +/-10% sous 250 mA environ pour alimenter la carte de programmation. Un bloc secteur style "prise de courant" s'il semble une solution séduisante et économique est cependant à éviter sauf si vous choisissez un modèle stabilisé. En effet, de nombreux blocs de ce type ne contiennent en fait qu'un transfo, un pont de diodes et un chimique et délivrent bien souvent une tension considérablement supérieure à celle marquée sur leur boîtier.

Il faut aussi prévoir un cordon série pour raccorder la carte de programmation à votre compatible PC. Cette dernière est munie d'une embase femelle 9 points type Sub D et respecte le brochage "normalisé" par les fabricants de PC.

L'assembleur MPALC

L'assembleur fourni avec le PICSTART est un produit de très bonne facture et ces compliments ne sont pas ceux d'un néophyte puisque l'auteur de ces lignes pratique la micro-informatique depuis plus de 15 ans! En effet, outre les possibilités classiques de tout assembleur, MPALC supporte la définition de macro-instructions ainsi que la notion d'assemblage conditionnel.

Il connaît bien évidemment les mnémoniques de tous les circuits de la série PIC 16CXX ainsi que les définitions classiques d'étiquettes, de constantes, de variables, etc... Les données numériques peuvent être exprimées dans toutes les bases classiques : binaire, décimal, octal, hexadécimal et caractère (ou ASCII si vous préférez).

Il autorise également l'écriture d'expressions arithmétiques, logiques et relationnelles au niveau des données, adresses et étiquettes et, là aussi, est particulièrement riche puisque l'on trouve par exemple le ou exclusif ou le décalage à droite ou à gauche ce qui est assez peu courant.

Le fichier à assembler doit lui être fourni en format texte ASCII pur et peut donc être produit avec n'importe quel éditeur de texte (EDIT de MSDOS par exemple) ou n'importe quel traitement de texte à condition de faire attention de demander une sauvegarde du fichier en ASCII. La **figure 1** montre ainsi un exemple de fichier texte à assembler, réalisé avec EDIT, et la **figure 2** le listing d'assemblage produit par MPALC.

La syntaxe à utiliser est classique pour un assembleur et se trouve même être

```
         TITLE "Sortie série du contenu d un registre"
         ;
         ; Début du programme
         ;
                  MOVLW 8            ; Charge 8 dans le registre W
                  MOVWF 11           ; Transfère 8 dans registre F11
         BOUCLE   BCF 6,1            ; Mise à zéro de l'horloge
                  RRF 0A,F           ; Rotation à droite de F10
                  BTFSS 3,0          ; Test de la retenue (F3, bit 0)
                  GOTO RETENZ        ; Saut si C=0
                  BSF 6,0            ; C=1, sortie d'un niveau haut
                  GOTO HORLOGE
         RETENZ   BCF 6,0            ; C=0, sortie d'un niveau bas
         HORLOGE  BSF 6,1            ; Mise à un de l'horloge
                  DECFSZ 11          ; Les 8 bits sont sortis ?
                  GOTO BOUCLE        ; Non
                  BCF 6,1            ; Oui, mise à zéro de l'horloge
                  END
```

Figure 1 : exemple type de listing source produit avec EDIT de MSDOS.

```
16c5x/7x Cross-Assembler V4.12 Released  Mon Dec 13 17:20:30 1993  Page 1
Sortie série du contenu d un registre

Line   PC    Opcode

0001                   ;
0002                   ; Début du programme
0003                   ;
0004   0000  0C08               MOVLW 8        ; Charge 8 dans le registre W
0005   0001  0031               MOVWF 11       ; Transfère 8 dans registre F11
0006   0002  0426  BOUCLE       BCF 6,1        ; Mise à zéro de l'horloge
0007   0003  032A               RRF 0A,F       ; Rotation à droite de F10
0008   0004  0703               BTFSS 3,0      ; Test de la retenue (F3, bit 0)
0009   0005  0A08               GOTO RETENZ    ; Saut si C=0
0010   0006  0506               BSF 6,0        ; C=1, sortie d'un niveau haut
0011   0007  0A09               GOTO HORLOGE
0012   0008  0406  RETENZ       BCF 6,0        ; C=0, sortie d'un niveau bas
0013   0009  0526  HORLOGE      BSF 6,1        ; Mise à un de l'horloge
0014   000A  02F1               DECFSZ 11      ; Les 8 bits sont sortis ?
0015   000B  0A02               GOTO BOUCLE    ; Non
0016   000C  0426               BCF 6,1        ; Oui, mise à zéro de l'horloge
0017   000D  0000               END

16c5x/7x Cross-Assembler V4.12 Released  Mon Dec 13 17:20:30 1993  Page 2


Cross-Reference Listing
LABEL       VALUE       DEFN      REFERENCES
BOUCLE      2           6         6      15
HORLOGE     9           13        11     13
RETENZ      8           12        9      12
```

Figure 2 : le listing d'assemblage fourni par MPALC.

```
SAMPLE              RADIX=X   MPSIM 4.11     16c55    TIME=0.00µ 0
------------------------------------------------------------------------
mulcnd: 00  mulplr: 00  H_byte: 00  L_byte: 00  count: 00  portb: 00  RB7: 1
RB6: 1  RB5: 1  RB4: 1  RB3: 1  RB2: 1  RB1: 1  RB0: 1

========================================================================
% AD count
% AD portb
% AD RB7,B,1
% AD RB6,B,1
% AD RB5,B,1
% AD RB4,B,1
% AD RB3,B,1
% AD RB2,B,1
% AD RB1,B,1
% AD RB0,B,1
% BC
% B main
% B call_m
% B mpy_S
% b loop
341216 bytes memory free
%
```

Figure 3 : l'affichage écran du simulateur : en haut la fenêtre de visualisation, en bas la fenêtre des commandes.

```
SAMPLE              RADIX=X   MPSIM 4.11     16c55    TIME=32.00µ 14
------------------------------------------------------------------------
mulcnd: 05  mulplr: 09  H_byte: 00  L_byte: 00  count: 08  portb: 05  RB7: 0
RB6: 0  RB5: 0  RB4: 0  RB3: 0  RB2: 1  RB1: 0  RB0: 1

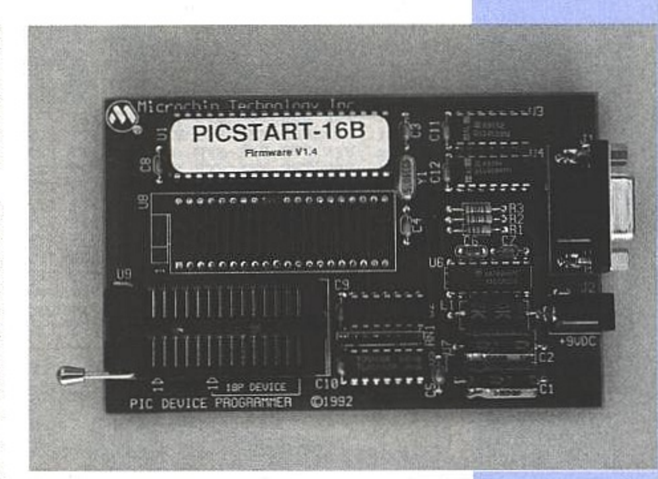
========================================================================
Processor Reset
% tc 0E
01FF 0A0E          goto    start                 |4.00µ 1  |
000E 0040 start    clrw                          |6.00µ 2  |W:0 F3:1C
000F 0002          option                        |8.00µ 3  |OPT:C0
0010 0206 main     movf    portb,w               |10.00µ 4 |W:9 F3:18
0011 0030          movwf   mulplr          ; m   |12.00µ 5 |F10:9 F3:18
0012 0206          movf    portb,w               |14.00µ 6 |W:5 F3:18
0013 0029          movwf   mulcnd                |16.00µ 7 |F9:5 F3:18
0014 0900 call_m   call    mpy_S           ; T   |20.00µ 8 |[15,0]
0000 0072 mpy_S    clrf    H_byte                |22.00µ 9 |F12:0 F3:1C
0001 0073          clrf    L_byte                |24.00µ 10 |F13:0 F3:1C
0002 0C08          movlw   8                     |26.00µ 11 |W:8
0003 0034          movwf   count                 |28.00µ 12 |F14:8 F3:1C
0004 0209          movf    mulcnd,w              |30.00µ 13 |W:5 F3:18
0005 0403          bcf     STATUS,CARRY    ; C   |32.00µ 14 |STATUS:18
%
```

Figure 4 : exécution d'un programme en mode trace.

moins contraignante que chez nombre de ses homologues. La tabulation peut être utilisée pour formater proprement le listing source sans que cela pose ensuite de problème lors de l'assemblage. Les étiquettes peuvent comporter jusqu'à 32 caractères et savent faire la distinction entre majuscules et minuscules encore qu'il soit possible d'interdire cette possibilité si on le désire.
En sortie, cet assembleur produit divers fichiers : un fichier objet bien sûr destiné à la programmation du micro, un fichier listing, un fichier des messages d'erreur et d'avertissement et un fichier de symboles destiné à être utilisé ensuite par le simulateur.
Différents formats de fichiers objets sont disponibles afin d'être compatibles non seulement du programmateur livré mais aussi des références du marché que sont les Data I/O et consort.
Cet assembleur supporte les macro-instructions ce qui lui confère une puissance accrue. Qu'est ce qu'une macro-instruction ? C'est un morceau de programme de votre choix, qui réalise généralement une fonction bien définie et que vous pouvez décider de baptiser d'un nom d'instruction qui vous est propre. Une fois définie de la sorte, cette macro-instruction peut être placée à tout endroit d'un programme et être reconnue correctement par l'assembleur. Pour mériter vraiment le nom de macro-instruction, elle doit supporter le passage de paramètres et la notion de variables locales ce qui est le cas ici.
L'assemblage conditionnel enfin est une des dernières possibilités de ce produit. Il permet, en fonction de la valeur de certains paramètres définis par vos soins, d'assembler telle ou telle partie d'un programme. C'est une possibilité très intéressante quoique méconnue de nombreux programmeurs. Elle permet d'écrire un programme général, valable pour plusieurs applications qui diffèrent seulement par des détails, par exemple un timer avec afficheur LCD ou afficheur à LED. Lors de l'assemblage, un ou plusieurs paramètres sont définis au début du listing pour dire à quelle application est destiné le programme. L'assembleur, grâce à la notion d'assemblage conditionnel, va alors choisir en fonction de l'état de ces paramètres les "morceaux" de programmes qui conviennent dans le listing général.

Le simulateur MPSIM

Le simulateur, à ne pas confondre avec un émulateur (revoir si nécessaire notre précédent article pour les précisions relatives à ces notions), permet de vérifier sur le compatible PC l'essentiel du comportement de votre programme.
Il simule bien évidemment le fonctionnement de l'unité centrale de l'un quelconque des circuits PIC 16CXX mais il supporte également la simulation des entrées/sorties au moyen de commandes particulières.
Même si son affichage en mode texte fait un peu "tristounet" à l'époque des logiciels à la sauce Windows comme le montre la **figure 3**, il est efficace pour ce qu'on attend de lui. En effet, on trouve en partie haute une fenêtre qui montre ce que l'on veut en ce sens que l'on peut définir soi même son contenu.
Il est ainsi possible d'y visualiser le contenu de registres, de mémoires particulières ou de ports d'entrées/sorties. La définition de ces éléments se fait en utilisant leurs noms normalisés ainsi que les étiquettes que vous leur avez affectées dans le programme à simuler. C'est là une possibilité très confortable puisqu'il n'est plus nécessaire de se souvenir de la correspondance étiquette - adresse. Ainsi, on voit sur cet exemple que l'on visualise un donnée appelée "mulcnd" dont on n'a pas à se soucier de l'adresse. On voit également que l'on peut y "éclater" la représentation d'un port comme c'est le cas pour le port B que l'on représente bit par bit avec RB7 à RB0. Les données de cette fenêtre évoluent bien évidemment au fur et à mesure du déroulement du programme simulé.
La partie basse de l'écran est la fenêtre de travail dans laquelle on frappe les commandes et où l'on observe leur exécution. Ainsi, la **figure 4** montre un exemple de fonctionnement en mode trace c'est à dire exécution pas à pas de N instructions. Notez que les temps théoriques d'exécution sont indiqués au niveau de chaque instruction suivis du numéro de pas du programme ainsi que de l'état des registres concernés.
Ce simulateur dispose de toutes les commandes classiques des produits de ce type: pose de points d'arrêt, exécution de programmes en mode pas à pas, exécution en mode trace avec diverses conditions d'arrêt, visualisation de la mémoire et des registres, etc..
Il permet de retoucher "à la main" un programme en cours de test et dispose d'une fonction permettant ensuite de lister les modifications ainsi faites de façon à pouvoir facilement les reporter dans le programme original.
Pour ce qui est des entrées/sorties, et comme sur tous les simulateurs de ce type, elles sont représentées en fait par des données lues et écrites en mémoire. Ici par contre il faut signaler une possibilité originale qui est celle du fichier de stimuli. Un tel fichier, tel celui présenté en exemple **figure 5**, contient les données à appliquer sur les pattes de tel ou tel port à des pas bien précis du programme. Ainsi, dans l'exemple choisi, lors du pas 3 de l'exécution du programme on applique 00001001 (9 en décimal) sur les lignes RB7 à RB0.
Pour les simulations répétitives d'un programme que l'on fait évoluer, il est vite fastidieux de définir toujours les mêmes conditions de départ (données affichées dans la fenêtre de visualisation, adresses des points d'arrêt, etc..). MPSIM supporte la notion de fichier de commande tel celui listé **figure 6** qui permet par son simple appel de faire exécuter autant de commandes que vous le désirez. Dans cet exemple on réalise ainsi la mise en place des différents éléments visualisés dans la fe-



La carte de programmation à connecter sur un port série.

```
! Stimulus file for SAMPLE.ASM

STEP      RB7   RB6   RB5   RB4   RB3   RB2   RB1   RB0   ! PortB Pins
3          0     0     0     0     1     0     0     1    !  9 x 5
5          0     0     0     0     0     1     0     1
65         0     0     0     0     1     0     1     0    ! 10 x 5
67         0     0     0     0     0     1     0     1
127        0     0     0     1     1     0     1     1    ! 27 x 3
129        0     0     0     0     0     0     1     1
191        0     0     0     1     0     0     0     1    ! 17 x 7
193        0     0     0     0     0     1     1     1
253        0     1     0     0     0     0     0     0    ! 64 x 63
255        0     0     1     1     1     1     1     1
```

Figure 5 : exemple de contenu d'un fichier de stimuli.

```
LO SAMPLE INHX8M
ST SAMPLE
NV
AD mulcnd
AD mulplr
AD H_byte
AD L_byte
AD count
AD portb
AD RB7,B,1
AD RB6,B,1
AD RB5,B,1
AD RB4,B,1
AD RB3,B,1
AD RB2,B,1
AD RB1,B,1
AD RB0,B,1
BC
B main
B call_m
B mpy_S
b loop
```

Figure 6 : un fichier de commande type facilitant l'exécution de simulations répétitives du même programme.



```
PicStart  File  Windows  Options  Device  Transfer  Config  P16C54   21:42:58
 ──────────────── C:\MICCHIP\SAMPLE.OBJ ────────────────   ┌[•] Fuses ════
 0000:   072 073 C08 034 209 403 330 603 rs.4..0.
 0008:   1F2 332 333 2F4 A06 800 040 002 .23...@.          Osc: XT
 0010:   206 030 206 029 900 A10 FFF FFF .0.)....          WDT: On
 0018:   FFF FFF FFF FFF FFF FFF FFF FFF ........          CP : On
 0020:   FFF FFF FFF FFF FFF FFF FFF FFF ........          ID : 0DFF
 0028:   FFF FFF FFF FFF FFF FFF FFF FFF ........          CkSum: 1D39
 0030:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0038:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0040:   FFF FFF FFF FFF FFF FFF FFF FFF ........           Id Edit
 0048:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0050:   FFF FFF FFF FFF FFF FFF FFF FFF ........          Fuse Edit
 0058:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0060:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0068:   FFF FFF FFF FFF FFF FFF FFF FFF ........
 0070:   FFF FFF FFF FFF FFF FFF FFF FFF ........

 F1 Help  Alt-X Exit    F4 Edit  F5 Program  F6 Verify  F7 Blank  F8 Read
```

Figure 7 : l'écran de contrôle du programmateur. La fenêtre de gauche montre ce qui va être programmé dans le circuit choisi.

nêtre haute puis la pose de points d'arrêt en diverses adresses définies seulement par leurs étiquettes (B main par exemple).

Précisons que tous les fichiers utilisés par ce simulateur sont des fichiers textes ASCII standards et que l'on peut donc les visualiser et les manipuler très facilement avec toutes les commandes du DOS.

Concluons cette présentation nécessairement brève en ajoutant que l'on peut faire sortir les résultats de simulation dans un fichier ou sur imprimante, que l'on peut réaliser un journal des simulations, que l'on peut désassembler du code, que l'on peut rechercher des données ou du code en mémoire, etc...

Bref, malgré sa présentation un peu aride, MPSIM est un simulateur efficace et performant qui mérite que l'on fasse l'effort de s'y intéresser.

Le programmateur

Dernier élément de la chaîne de production, le logiciel de programmation est le plus simple des trois. Il s'agit en effet d'un programme à présentation pseudo-graphique avec menus déroulants et boites de dialogue comme le montre à titre d'exemple la recopie d'écran de la **figure 7**.

Il permet les manipulations classiques sur tous les programmateurs de ce type: sélection du circuit à programmer, test de virginité, programmation, vérification, plus un certain nombre de fonctions annexes moins importantes.

Il fonctionne avec la carte fournie qui peut être connectée sur le port 1 à 4 du compatible PC et dont il teste le bon fonctionnement de la liaison avant toute tentative de programmation.

Son fonctionnement n'appelle pas de commentaire particulier, pas plus que celui de la carte qui, même si elle est munie d'un support à force d'insertion nulle, n'est tout de même pas un programmateur de production capable de "sortir" des centaines de pièces par jour.

Conclusion

Le PICSTART 16 B n'est certes pas parfait, ses plus grosses lacunes se situant au niveau de la documentation qui comporte quelques erreurs (essayez de faire l'exercice proposé en guise de prise en main du simulateur tel qu'il est décrit et nous vous souhaitons bien du plaisir !). On peut aussi lui reprocher de ne fournir les modes d'emploi de l'assembleur et du simulateur que sous forme de fichiers textes à faire imprimer et de nécessiter l'achat d'une petite alimentation et d'un câble série pour PC.

En contrepartie, les programmes fournis, assembleur et simulateur en particulier, sont d'un excellent niveau technique et permettent de mener à bien le développement de nombreuses applications avec un confort honnête.

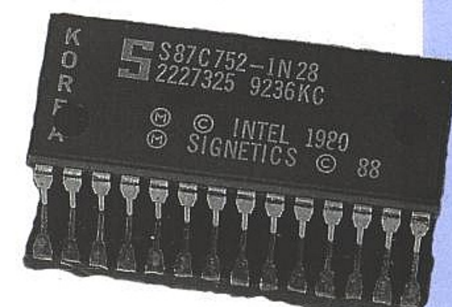
Ceci permet d'oublier bien vite les petites critiques précédentes, surtout eu égard au prix de vente du produit que nous n'hésiterons pas à qualifier de très intéressant.

C.BROUSSAS

DES MINI « 80C51 » : LES 8X C751 ET C752

Comme vous avez pu le constater depuis plusieurs numéros d'ERP, de petits microcontrôleurs, ST6, PIC, ... ont fait des entrées remarquées sur le marché et font fureur pour des applications simples et économiques.





Loin d'être en retard dans ce domaine, la famille 8x C 51 possède depuis déjà plus de trois ans des « petits » microcontrôleurs performants et économiques dont le gros avantage réside dans le fait de ne pas nécessiter de nouveaux investissements (en matériel de développement et connaissances - apprentissage), autres que ceux déjà effectués pour cette famille largement connue et diffusée. Les trois principaux petits dérivés de cette famille sont par ordre croissant de performances, les 8x C750, C751, C752.

Le tableau de la **figure 1** résume leurs performances respectives.
Comme vous avez certainement pu le remarquer les deux derniers types disposent d'une interface I2C.
En ce qui concernera les applications proposées, nous vous donnerons le canevas pour réaliser Codeur/Décodeur de télécommandes IR (RC5) interfacées ou non I2C, des applications d'ACCESSoires pour PC (souris) (voir série d'articles sur l'access-bus), un répertoire téléphonique de poche à génération de tonalités DTMF, une interface UART/I2C, etc.
La plupart de ces applications utiliseront le bus I2C pour perpétuer la souplesse de conception qu'offre ce bus à ses utilisateurs.
Pour des raisons de coût global du composant, l'interface intégrée (hardware) I2C des 8x C751 et C752 a été conçue de façon différente (gestion bit à bit) de celles des 8x C652, C654, C552 (gestion byte à byte) que nous avons décrites précédemment. Aussi, avant toute chose il est nécessaire d'examiner son principe de fonctionnement en détails.

L'interface Hardware des 8x C751 et C752 ou l'I2C orienté « BIT »

Pourquoi vous entraîner autour des 8x C751 ou C752. C'est bien simple ces « petits » microcontrôleurs, à cœur de 80 C51, sont peu chers, très performants, disponibles en versions Rommées, OTP et très employés dans les réseaux ACCESS-bus.

Bien que ne possèdant que des ressources réduites comparativement à leurs grands frères (voir **figure 2**) mais grâce à leurs spécificités (convertisseurs D/A et PWM pour le C752...) ils sont largement suffisants pour satisfaire la plupart des applications d'ACCESSoires envisagées (souris, ligth-pen, etc.) dans le cadre de réseaux ACCESS-bus, les 80 C652, C654, C552 plus performants étant plus fréquemment rencontrés sur les cartes-mère CPU.

Les SFR's I2C du 8x C751

Comme vous avez pu le remarquer, ces microcontrôleurs sont dérivés du 8x C51 ne possèdent que peu de mé-

80CXXX = ROM-Less 83CXXX = Masked-ROM 87CXXX = EPROM/OTP 89CXX = Flash EEPROM

DEVICE	OTP & EPROM	ROM (bytes)	RAM (bytes)	MAX SPEED (MHz)	8-BIT PORTS	SERIAL I/O UART	SERIAL I/O I2C	TIMERS	A/D BITS	A/D CHANNELS	SPECIAL FEATURES	PACKAGES
8XC750	1K	-	64	40	2+3/8			1			40 MHz, 24-Pin Skinny DIP Package	A28, F24, N24
8XC751	2K	2K	64	16	2+3/8		•	1			24-Pin Skinny DIP Package	A28, F24, N24
8XC752	2K	2K	64	16	2+5/8		•	1	8		PWM	A28, F28, N28

Figure 1



moire ROM et RAM et ont donc des cristaux de faibles dimensions.
Comme nous vous l'avons laissé entendre lors des paragraphes précédents, afin de ne pas augmenter démesurément la surface et le coût du composant, c'est l'interface hardware « orienté bit » qui a été retenu pour assurer la liaison I2C. Cela demandera donc une gymnastique logicielle un peu plus importante pour faire fonctionner le composant en mode multimaster.

Description de l'interface « orientée bit » des 8x C751 et C752

Comme précédemment, c'est par l'intermédiaire des SFR's de la CPU du microcontrôleur que cela va se passer.
La **figure 3**, indique le mapping des SFR's du C751 (identiques à ceux des C752 en ce qui concerne le bus I2C).
Malgré un air de famille avec les appellations précédentes, ils sont très sensiblement différents et méritent une attention particulière (hélas pour vous comme pour nous !) « bit à bit ».

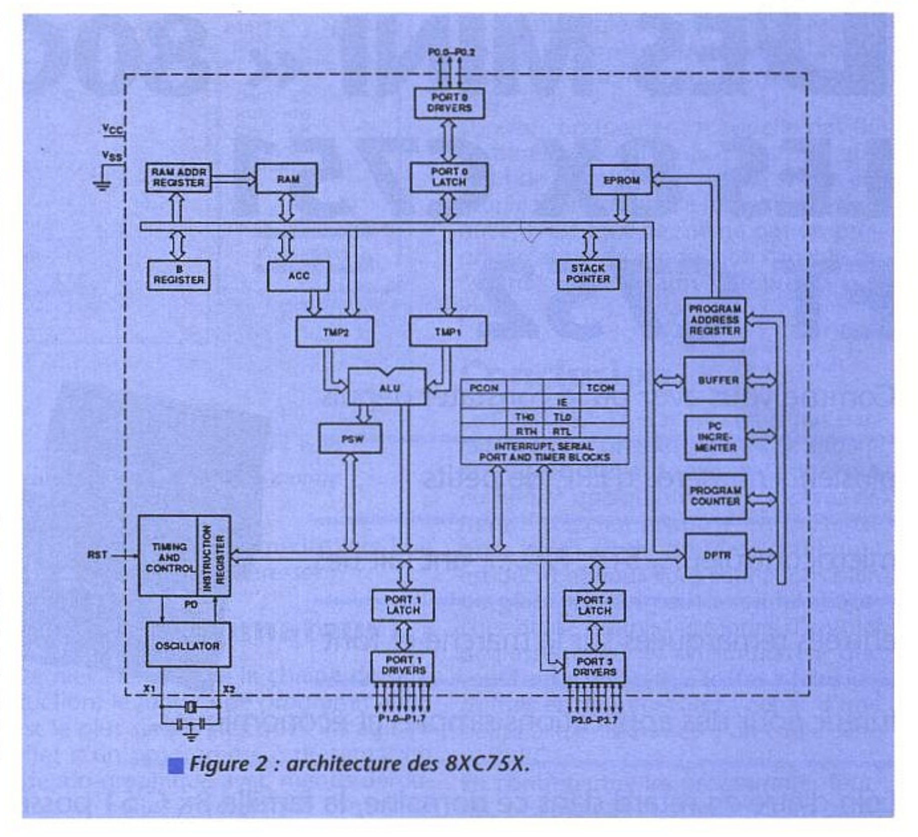


Figure 2 : architecture des 8XC75X.

Structure générale de l'interface I2C

Pour tenter de mieux comprendre, revenons un instant sur la signification « interface hardware orientée bit »
Ceci signifie que de façon « hardware », seules les portes nécessaires à des procédures d'arbitrage, d'erreur de format de trame, d'extraction de l'horloge, de signification et traitement des erreurs de débordement temporel du bus ont été implantées et que le traitement des signaux qui seront (ou qui devront être) présents sur le bus I2C seront traités électriquement par un système d'interruption « bit après bit » et non pas octet après octet.
L'interface est donc synchronisée par le logiciel soit au travers de « boucles de scrutation » soit par « interruptions ».
Ceci implique donc certaines contraintes au niveau du fonctionnement demandé au microcontrôleur et quelques particularités que nous allons maintenant vous décrire.

fonctionnement de l'interface I2C du 8x C751

Le Timer 1 est utilisé à deux fins :
- d'une part pour la commande des timings du bus I2C
- d'autre part en « chien de garde » afin de détecter un éventuel blocage parasite du bus et à cette occasion savoir seul déclenché automatiquement une interruption si rien ne s'est produit sur les lignes SDA, SCL du bus I2C pendant un temps relativement long.
Si l'interruption ne se produit pas, le programme peut permettre de corriger l'erreur et autoriser de répéter la dernière transmission I2C.
Pour bien comprendre comment fonctionne cette interface, il faut savoir que six tranches de temps sont importantes dans le fonctionnement du bus I2C et qu'elles sont assurées par le Timer 1 :
Lorsque le microcontrôleur est maître :
- le temps minimum où SCL est haut ;
- le temps minimum où SCL est bas ;
puis d'une façon générale :
- Les trois conditions temporelles pendant les phases de STOP et de START du bus I2C qui sont les temps minimum de :
• SCL Haut à SDA haut pour le STOP
• SDA Haut à SDA bas pour le STOP et le START
• SDA Bas à SDA Haut pour le START.
- Enfin, la valeur du temps réprésentatif de la vitesse maximale de changement des signaux de l'horloge (SCL) lors d'un échange I2C dont il est aisé de se servir pour détecter et indiquer soit des erreurs dues à des problèmes de réponses logicielles, soit la présence de composants défectueux ou encore des parasites (ou bruit) sur le bus dus à d'autres maîtres tentant de prendre le bus lors de phase de synchronisation et/ou d'arbitrage.
Les cinq premières valeurs sont égales à 4,7 µs dans la spécification (standard) du bus I2C et, étant donné que le Timer T1 du 8xC 751 est « clocké » à la fréquence horloge du microcontrôleur (pouvant être choisie entre 0,5 et 16 MHz), il est possible de précharger le rapport de division de celui-ci à l'aide des deux derniers bits d'un registre (CT1 et CT0, registre I2CFG, en l'occurence) de façon à optimiser et ne pas dégrader les temps de réponse logiciel en présence d'une fréquence d'horloge faible.
En ce qui concerne le dernier paramètre, il est important de le respecter mais sa valeur n'est pas critique. A cet effet la totalité des 10 bits du compteur du Timer de T1 est exploitée pour définir la valeur maximale de ce paramètre.
Son fonctionnement est le suivant :
- ce compteur n'est actif que pendant la présence d'une trame I2C ;
- lorsque l'I2C est opérationnel, le compteur est remis à zéro à chaque transition présente sur la broche SCL ;
- lorsqu'il est actif, il compte jusqu'à 1022 (ou 1023) cycles machines après la dernière transition et, si la retenue a lieu, elle provoque d'une part un reset hardware de l'interface I2C du circuit intégré 8x C751, et d'autre part elle génère une interruption si celle-ci est autorisée.
Dans les cas où le bus est bloqué (à cause par exemple d'une faute de réponse logicielle), le reset libère le SCL et permet au bus I2C de continuer de fonctionner avec un autre circuit un peu moins recalcitrant.

Les interruptions I2C

Si les interruptions I2C sont autorisées (les bits EA et EI2 étant tous deux positionnés à « 1 »), une interruption I2C peut se produire chaque fois que le bit « ATN » sera positionné par une condition de START, de STOP, perte d'arbitrage, ou une condition de « donnée prête » (« DRDY » - data ready).
En pratique il n'est pas souhaitable de faire fonctionner l'interface I2C dans ce mode car la routine de service d'interruption devra distinguer son chemin quelque part parmi les centaines de possibilités pouvant se produire.
Aussi pour que l'I2C puisse fonctionner à une vitesse plus élevée, il est préférable que le logiciel s'exécute plus vite sur ordre de l'interface I2C lui-même.
Normalement l'interruption I2C ne devrait servir qu'à indiquer une condition de départ à un Esclave en sommeil (idle) ou une condition de STOP à un Maître en sommeil (idle) (si il était en position d'attente d'utiliser le bus I2C).
A noter que tout ceci est facilement



réalisable en n'autorisant les interruptions que pendant les conditions ci-dessus.

Les SFR's I2C du 8x C751

Etudions maintenant les SFRs spécifiques du 8x C751 concernant l'I2C. Au nombre de quatre, ils vont servir à faire fonctionner l'interface hardware de ce microcontrôleur.
Il s'agit des :
I2CON pour « I2C control » (en bon français commande de l'I2C) ;
I2CFG pour « I2C configuration » ;
I2DAT pour « I2C data » (données) ;
I2STA pour « I2C status »
voir **figure 4** et **figures 5, 6, 7, 8.**
Ces registres sont très malicieux et méritent un examen complet car le fonctionnement bit à bit de l'interface peut rendre acrobatique la mise en route de l'interface et de son logiciel associé. Prenez maintenant votre courage à deux mains et commençons l'examen de leur contenu bit à bit... et dans l'ordre SVP !

Description des registres

Le registre I2CON
La figure 6, indique son contenu qui est double selon qu'on le lit bien qu'on l'écrit.
à la lecture :
* RDAT
la valeur du bit de « données » présent sur le fil SDA est capturée et est introduite dans le bit « Receive DATa » au moment où le flanc de montée a lieu sur le signal SCL (à noter que la valeur de cette même donnée est aussi présente dans le bit de poids le plus fort du registre I2DAT en lecture, agrémentée de 7 zéros d'accompagnement).
Pourquoi donc faire simple quand on peut faire compliqué si facilement.
Le fait d'aller lire la valeur du bit RDAT dans le registre I2DAT nettoie en même temps le bit DRDY (data ReaDY) permettant à l'interface hardware I2C de traiter l'arrivée d'un autre bit.
Dans une application standard, les sept premiers bits d'un message sont lus partir du IDAT et le dernier (le 8e) à partir du I2CON. (Pourquoi donc ? un peu de patience !) puis, I2DAT peut être écrit pour envoyer le bit d'ACK et nettoyer le bit DRDY.
* ATN
De son vrai nom « ATteNtion », ce bit est à 1 lorsque un ou plusieurs des bits suivant : DRDY, ARL, STR, ou STP sont à 1.
Ce bit peut être testé pour libérer la routine de service I2C lors « d'une boucle d'attente ».
* DRDY
Le bit Data ReaDY (et donc ATN) est positionné à 1 lorsqu'un flanc de montée est présent sur SCL (sauf lorsque le microcontrôleur est en mode idle et de plus en Esclave I2C).
Le bit DRDY est nettoyé (remis à zéro) en écrivant un 1 dans son alter ego le bit CDR (Clear Data Ready) ou en écrivant ou lisant le registre I2DAT.
L'état bas suivant de SCL est extrait jusqu'à ce que le programme réponde en nettoyant le bit DRDY.
Petite remarque
Si un programme détecte ATN = 1, il devrait normalement vérifier par la suite la valeur de DRDY.
Si DRDY = 1, alors, et s'il reçoit le dernier bit, il devrait capturer la valeur de la donnée à partir de RDAT (dans I2DAT ou I2CON). Puis, le bit suivant doit être transmis, et devrait être écrit dans I2DAT.
D'une façon ou d'une autre, il devrait nettoyer DRDY et retourner surveiller ATN.
* ARL
La perte d'arbitrage est signalée par le bit ARL (arbitration lost).
Ce bit passe à 1 quand le bit « transmit active » (registre I2DAT) est positionné. De même le bit Transmit Acive est nettoyé quand ARL passe à 1.
Il y a quatre raisons différentes pour lesquelles le bit ARL passe à 1 :
a)
si le programme envoie un « 1 » ou un restart mais un autre élément envoie un « 0 » ou un stop tel que SDA soit à l'état bas lors de la montée de l'horloge SCL.
b)
si le programme envoie un « 1 » mais un autre élément envoie un restart, il amène SDA est à l'état bas avant que le microcontrôleur puisse amener SCL à l'état bas.

SYMBOL	DESCRIPTION	DIRECT ADDRESS	MSB							LSB	RESET VALUE
			BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	High byte	83H									00H
DPL	Low byte	82H									00H
			DF	DE	DD	DC	DB	DA	D9	D8	
I^2CFG*#	I^2C configuration	D8H/RD	SLAVEN	MASTRQ	0	TIRUN	–	–	CT1	CT0	0000xx00B
		WR	SLAVEN	MASTRQ	CLRTI	TIRUN	–	–	CT1	CT0	
			9F	9E	9D	9C	9B	9A	99	98	
I^2CON*#	I^2C control	98H/RD	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	–	81H
		WR	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP	
I^2DAT*#	I^2C data	99H/RD	RDAT	0	0	0	0	0	0	0	80H
		WR	XDAT	X	X	X	X	X	X	X	
			FF	FE	FD	FC	FB	FA	F9	F8	
I^2STA*#	I^2C control	F8H	–	IDLE	XDATA	XACTV	MAKSTR	MAKSTP	XSTR	XSTP	x0100000B
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*#	Interrupt enable	A8H	EA	–	–	EI2	ETI	EX1	ET0	EX0	00H
								82	81	80	
P0*#	Port 0	80H	–	–	–	–	–	–	SDA	SCL	xxxxx111B
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	T0	$\overline{INT1}$	$\overline{INT0}$	–	–	–	–	–	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON#	Power control	87H	–	–	–	–	–	–	PD	IDL	xxxxxx00B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*#	Timer/counter control	88H	GATE	C/T	TF	TR	IE0	IT0	IE1	IT1	00H
TL#	Timer low byte	8AH									00H
TH#	Timer high byte	8CH									00H
RTL#	Timer low reload	8BH									00H
RTH#	Timer high reload	8DH									00H

Figure 3 : mapping des registres SFR

REGISTER ADDRESS			BIT ADDRESS							
NAME	SYMBOL	ADDRESS	MSB							LSB
I^2C control	I2CON	98	9F	9E	9D	9C	9B	9A	99	98
I^2C data	I2DAT	99	–	–	–	–	–	–	–	–
I^2C configuration	I2CFG	D8	DF	DE	DD	DC	DB	DA	D9	D8
I^2C status	I2STA	F8	FF	FE	FD	FC	FB	FA	F9	F8

Figure 4

	7	6	5	4	3	2	1	0
Read	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	–
Write	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP

Figure 5

	7	6	5	4	3	2	1	0
Read	SLAVEN	MASTRQ	0	TIRUN	–	–	CT1	CT0
Write	SLAVEN	MASTRQ	CLRTI	TIRUN	–	–	CT1	CT0

Figure 6

	7	6	5	4	3	2	1	0
Read	RDAT	0	0	0	0	0	0	0
Write	XDAT	X	X	X	X	X	X	X

Figure 7

Read only

7	6	5	4	3	2	1	0
–	IDLE	XDATA	XACTV	MAKSTR	MAKSTP	XSTR	XSTP
MSB							LSB

Figure 8



c)
en mode Maître si le programme envoie un restart mais un autre élément envoie un « 1 » et amène SCL à l'état bas avant que le 8x C751 puisse mettre à « 0 » la ligne SDA.
d)
en mode Maître si le programme envoie un stop mais qu'il ne peut pas réaliser car un autre élément envoie déjà un « 0 ».
* STR
le bit de « STaRt » est positionné à « 1 » lorsque une condition de start est détectée (soit le composant en Maître soit en Esclave non idle)
*STP
le bit de « SToP » est positionné à 1 lorsqu'une condition de stop est détectée sur le bus (plus les mêmes réserves).
* MASTER
ce bit à 1 indique que le 8x C751 est actuellement le Maître du bus I2C.
Ce bit est positionné quand le bit du registre de configuration MASTRQ (master request) est placé à « 1 » et que le bus est libre.
La remise à zéro de ce bit est effectuée quand ARL est mis à « 1 » ou bien quand on passe le bit MASRQ à « 0 » et qu'un stop a été effectué.
Vous pensiez en avoir terminé avec ce registre et bien non !
Passons maintenant à l'écriture de ce même registre.
I2CON en écriture :
Sur le principe, à chaque bit transmis sur l'I2C une routine de service devrait attendre que ATN passe à « 1 » pour toutes les raisons évoquées précédemment (DRDY, ARL, STR, STP) et, selon la position du bit de données dans le message transmis, il devrait écrire dans le registre I2C l'un des bits suivants (où il doit lire ou écrire le registre I2DAT).
* CXA
le fait d'écrire un « 1 » dans le bit « Clear Xmit Active » nettoie l'état « transmission active ».
(Notons à ce propos que le fait de lire I2DAT effectue aussi la même chose.)
La transmission est active dès lors que l'on écrit dans le registre I2DAT ou bien encore en positionnant dans I2CON les bits XSTR ou XSTP « 1 ».
* IDLE
Le fait d'écrire un « 1 » dans ce bit entraîne que le hardware I2C Esclave ignorera ce qu'il se passe sur le bus I2C jusqu'à la détection d'une nouvelle condition de START (si le bit de MASTRQ est à « 1 » alors une condition de stop fera passer le 8x C751 en mode Maître).
* CDR
Le fait d'écrire un « 1 » dans le bit « Clear Data Ready » nettoie le bit DRDY (notons à ce propos que le fait de lire ou d'écrire I2DAT effectue la même chose).
* CARL
Le fait d'écrire un « 1 » dans le bit « Clear Arbitration Lost » nettoie le bit ARL.
* CSTR
Le fait d'écrire un « 1 » dans le bit « Clear STarRt » nettoie le bit STR.
* CSTP
Le fait d'écrire un « 1 » dans le bit « Clear SToP » nettoie le bit STP. On

SLAVEN, MASTRQ, MASTER	TIRUN	OPERATING MODE
All 0	0	The I^2C interface is disabled. Timer I is cleared and does not run. This is the state assumed after a reset. If an I^2C application wants to ignore the I^2C at certain times, it should write SLAVEN, MASTRQ, and TIRUN all to zero.
All 0	1	The I^2C interface is disabled. Timer I operates as a free-running time base. Use this mode only in non-I^2C applications.
Any or all 1	0	The I^2C interface is enabled. The 3 low-order bits of Timer I run for min-time generation, but the hi-order bits do not, so that there is no checking for I^2C being "hung." This configuration can be used for very slow I^2C operation.
Any or all 1	1	The I^2C interface is enabled. Timer I runs during frames on the I^2C, and is cleared by transitions on SCL, and by Start and Stop conditions. This is the normal state for I^2C operation.

Figure 9

CT1, CT0	OSC/12 COUNT	f_{osc} MAX	TIMEOUT PERIOD
10	7	16.8MHz	1023 cycles
01	6	14.25MHz	1022 cycles
00	5	11.7MHz	1021 cycles
11	4	9.14MHz	1020 cycles

Figure 10

peut remarquer que si l'un ou plusieurs des bits DRDY, ARL, ou STP sont à 1, la période basse de SCL s'étirera jusqu'à ce que la routine de service réponde de les nettoyer.
* XSTR
Le fait d'écrire un « 1 » dans le bit « Xmit repeated STaRt » demande au hardware de générer un signal de restart sur le bus I2C.
* XSTP
Le fait d'écrire un « 1 » dans le bit « Xmit SToP » demande au hardware de générer une condition de stop sur les fils SDA et SCL du bus I2C.
Le registre I2DAT
Le contenu de ce registre est présenté sur la **figure 7**.
Ce registre fonctionne aussi en écriture et en lecture.
En lecture :
* RDAT
« Le » bit, l'unique bit d'où l'appelation « d'interface I2C bit à bit » est récupéré en réception dans le bit « Receive DATa ». En fait il est capturé sur la ligne SDA à chaque flanc de montée de la ligne SCL.
A remarquer que le fait de lire I2DAT nettoie DRDY et l'état de transmit active.
à l'écriture :
* XDAT
« Xmit Data » prépare la donnée pour le prochain bit.
Notons que le fait d'écrire I2DAT nettoie DRDY et positionne l'état de Transmit Active.
Le registre I2CFG
Le contenu de ce registre est présenté sur la **figure 6**.
Ce registre fonctionne lui aussi en écriture et en lecture.
Afin de vous éviter une trop longue tirade concernant ce registre, nous allons en quelques mots résumer l'essentiel des fonctions de configuration qu'il renferme. En effet ce registre est plein de fines subtilités qui dépassent largement le cadre de cette revue. Notre but actuel se résume à vous indiquer pour l'instant le fonctionnement général de cet interface afin que vous puissiez par la suite comprendre comment fonctionneront les différentes applications que nous vous proposerons.
* SLAVEN
positionne le composant en mode Esclave.
* MASTRQ
positionne le composant en mode Maître.
* CLRT1
Le fait d'écrire un « 1 » dans ce bit nettoie le drapeau de l'interruption du Timer 1.
* T1RUN
Le fait d'écrire un « 1 » dans ce bit lance le Timer 1. Un « 0 » l'arrête et le nettoie.
* CT1, 0
Ces deux bit permettent de choisir le débit du bus à l'aide de diviseurs internes.
Au lieu d'un long discours les tableaux **figures 9 et 10**, résument les actions et interactions de ces bits turbulents.
Le registre I2STA
La **figure 8** rappelle son contenu.
Ce registre ne peut être que lu. OUF !!
Il reflète l'état de l'interface hardware de l'I2C.
Les trois bits sont des doublons (en lecture) des bits IDLE, XSTR et XSTP que vous avez écrits dans le registre I2CON.
* XDATA
Contenu du buffer de transmission
* XACTV
Emetteur actif.
* MASKSTR
Ce bit est à « 1 » lorsque le circuit effectue une condition de start.
* MASKSTP
Ce bit est à « 1 »quand le circuit effectue une condition de stop.
* XSTR
Ce bit est à « 1 » quand le circuit effectue une condition de restart.
* XSTP
Ce bit est à « 1 » quand le circuit effectue une condition de restop.
Et pour terminer cette longue mais nécessaire lithanie, disons deux mots de l'interruption dédiée à l'I2C du microcontrôleur 8x C751.
Le bit de validation de l'interruption de l'I2C a pour nom EI2 que le vecteur d'interruption de l'I2C loge à l'adresse 023 de la mémoire programme en ayant la priorité la plus basse dans la hiérarchie des interruptions.

Evidemment ce n'est pas aussi simple qu'avec un interface byte à byte intelligente de type 8x C652 ou C552 mais la surface de silicium s'en ressent d'autant car la surface occupée par les octets en ROM est moins volumineuse que la logique cablée nécessaire dans l'autre cas.
Bref au prix de ses complications, le coût du composant est plus faible.

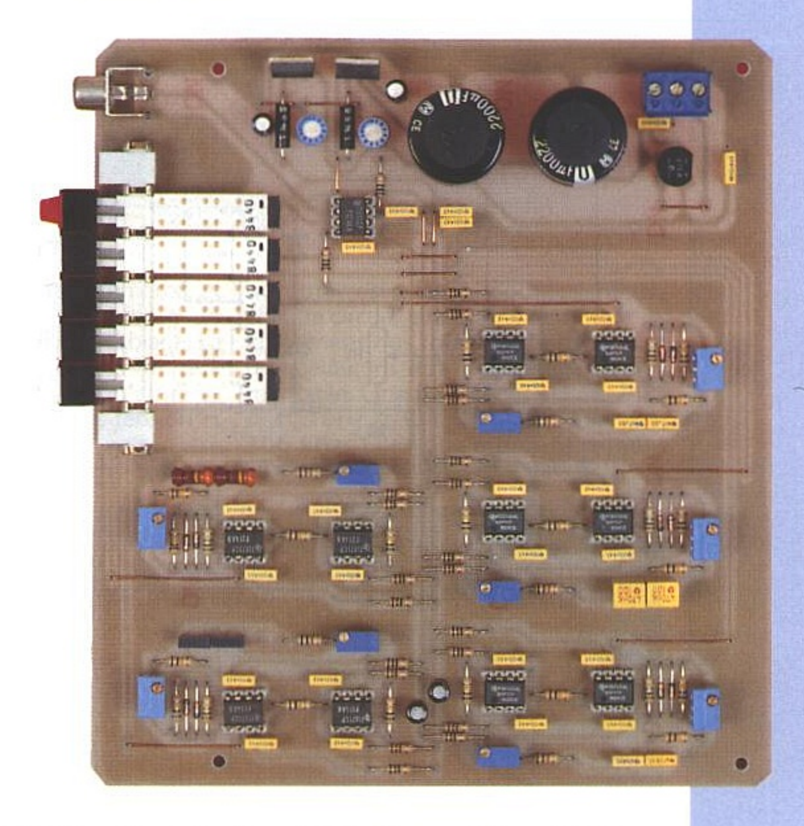
Nous vous donnons rendez-vous le mois prochain pour les premières applications de ces composants.
A noter que de nombreuses routines « standard » de gestion du bus I2C pour les 8x C751 et C752 se trouveront sur le serveur ERP.

D. PARET



OSCILLATEURS SINUSOÏDAUX À PONT DE WIEN

Il peut être intéressant et utile de disposer d'un générateur de sinusoïdes de fréquences fixes, stables et d'amplitude de sortie constante. Ce générateur pourra être utilisé, entre autre, pour le test ou l'alignement de matériel basse fréquence puisque les fréquences produites iront de 10Hz à 100kHz, ou encore pour effectuer des mesures sur les composants passifs.



L'oscillateur de Wien

C'est le moyen le plus simple pour obtenir des sinusoïdes. Le schéma de base est donné en **figure 1a**. Pour obtenir la naissance des oscillations, il faut que la résistance RA soit égale à deux RB, ce qui donne à l'ensemble un gain de 3. Mais tel qu'il est représenté sur cette figure, le système n'est pas stable :

1/ si le rapport des résistances RA et RB est exactement égal à 2, et si donc le gain de l'amplificateur opérationnel est exactement égal à 3 (puisque monté en amplificateur non inverseur), alors l'équilibre s'établit. Mais l'amplitude de sortie reste très faible, et de toute façon inutilisable ;

2/ si le rapport des résistances RA et RB est légèrement inférieur à 2, le gain inférieur à 3, le signal réinjecté est trop faible et les oscillations cessent ;

3/ si le rapport des résistances RA et RB est légèrement supérieur à 2, et donc ,le gain supérieur à trois, les oscillations commencées se perpétuent dans le temps, mais leur amplitude ne cessent d'augmenter sans qu'il soit possible de les stabiliser à une valeur donnée (divergence). Cette amplitude augmente jusqu'à l'écrêtage, et l'on obtient une sorte de signal rectangulaire. Il faut donc un système qui fera passer le gain de l'AOP à une valeur supérieure à trois afin que les oscillations prennent naissance, mais qui en même temps, diminuera ce gain par la suite pour ne pas engendrer de distorsions indésirables. La fréquence obtenue est égale à $F = 1 / 2\pi RC$. Divers dispositifs existent pour la stabilisation des oscillateurs. Le plus courant est le remplacement de la résistance RB par une lampe à filament de tungstène qui joue le rôle d'une résistance à coéfficient de température positif. Elle augmente

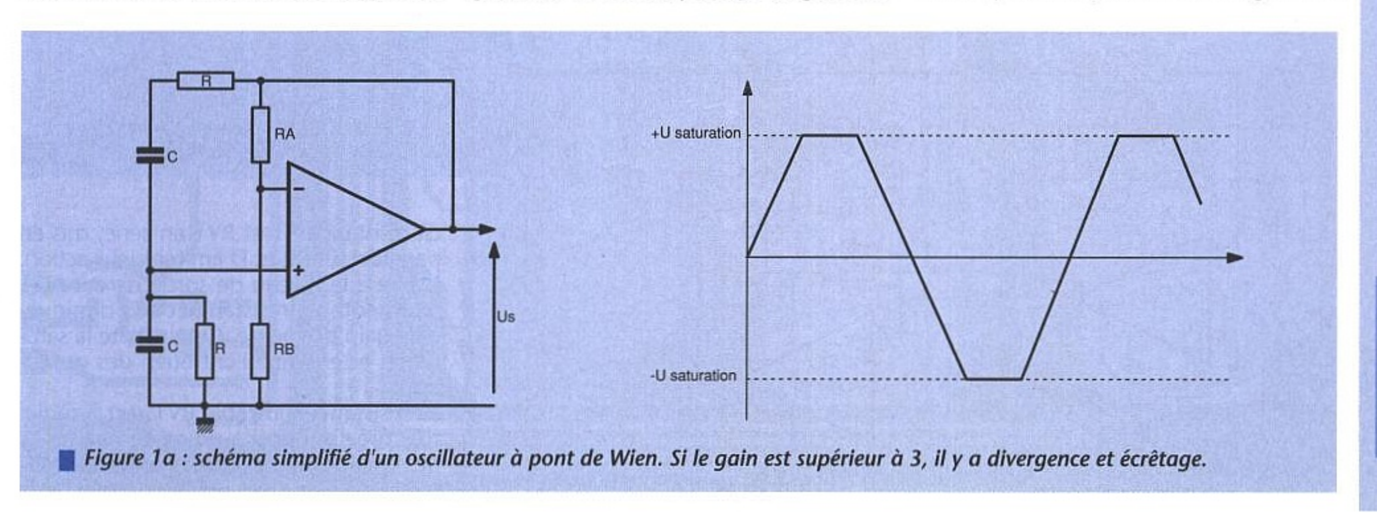


Figure 1a : schéma simplifié d'un oscillateur à pont de Wien. Si le gain est supérieur à 3, il y a divergence et écrêtage.

quand la tension de sortie croît diminuant ainsi le gain ; la tension de sortie diminue alors, la résistance également, et le gain repasse le seuil de 3, ce qui maintient le processus des oscillations. Deux diodes montées tête bêche en sortie de l'AOP et qui sont donc en série avec RA peuvent également jouer le rôle de résistance ajustable, puisque leur résistance dynamique varie suivant le courant qui les traverse.
Un autre moyen de stabilisation est donné en **figure 1b.** Là, la résistance RB est remplacée par un transistor à effet de champ, qui voit sa résistance varier selon la tension appliquée sur sa grille. A la sortie de l'amplificateur opérationnel, on trouve un redressement et un fitrage opéré par D et CA. A la connexion de ces deux composants, on trouve donc une tension négative par rapport à la masse. Le curseur du potentiomètre RC prélève une partie de cette tension qui est appliquée à la grille du FET. Plus cette tension sera élevée, plus la résistance drain - source sera élevée, et plus le gain sera faible. Nous avons donc là, un système de régulation de l'amplitude de sortie.
La **figure 1c** représente le générateur que nous avons employé pour la réalisation de nos oscillateurs sinusoïdaux. Il est tiré d'une note d'application de constructeur. Il est d'un fonctionnement sûr, et d'une stabilité en fréquence et en amplitude de sortie très acceptables.
Lorsque le niveau de sortie du deuxième amplificateur est proche de 0V, les diodes D présentent une grande résistance. Le gain est alors à son maximum.

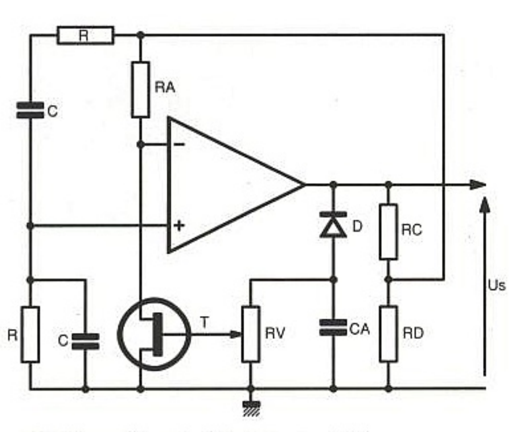


Figure 1b : stabilisation par FET.

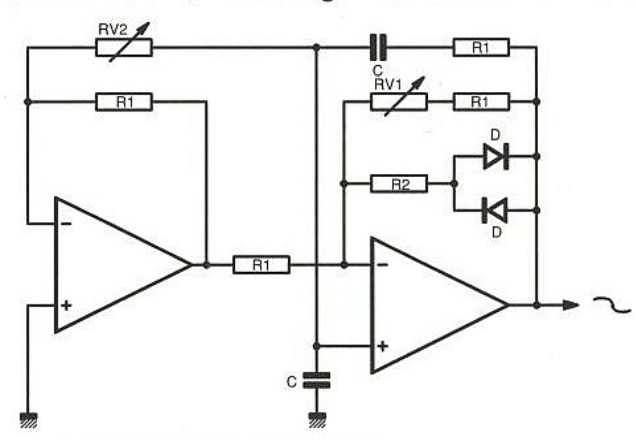
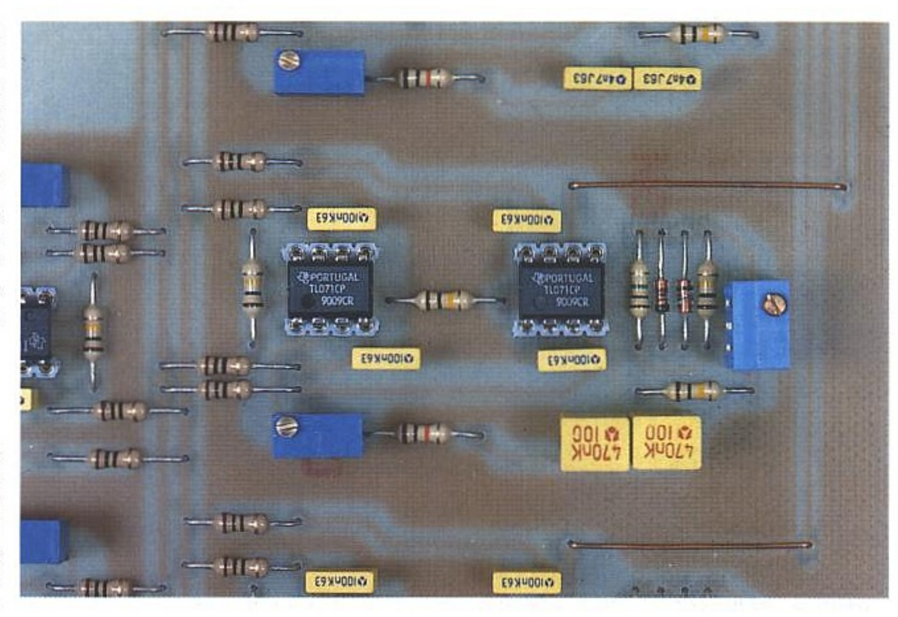


Figure 1c : stabilisation par diodes.

Figure 2 : le schéma complet des cinq oscillateurs commutés.

La résistance R1 et RV1 en série, mis en parallèle sur R2 et D limitent leur action. Lorsque le niveau de sortie de cet AOP augmente, la résistance de D diminue, et le gain de même. Ce qui évite la saturation et entraine l'entretien des oscillations.
La résistance ajustable RV1 sert à règler très précisément le point de naissance des oscillations, et donc l'amplitude de sortie des sinusoïdes. Elle permet l'ob-

tention d'un niveau allant de quelques millivolts à plusieurs volts, ce qui permettra le règlage à la valeur souhaitée. La résistance RV2 permet d'ajuster très précisément la fréquence des signaux, et ce, sur une faible plage.
La stabilité en fréquence et en tension d'un circuit de ce type ne sera obtenue qu'après un certain temps de fonctionnement. En effet, les diodes étant sensibles à la chaleur, une certaine instabilité (faible) sera observée au début de la mise sous tension.



LES OSCILLATEURS

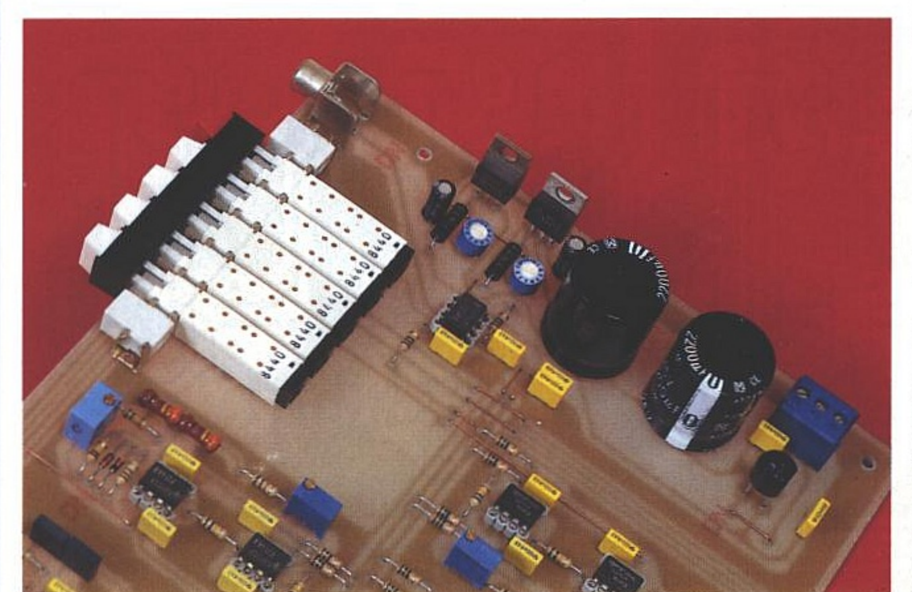
Le schéma de principe général est donné en **figure 2.** Comme on peut le constater à la vue de ce dessin, ce n'est pas d'un mais de cinq oscillateurs dont il est question, chacun se chargeant d'une fréquence. Une commutation des condensateurs et des résistances ajustables aurait pu être effectuée, ce qui aurait notablement simplifié le circuit imprimé et diminué le nombre des composants. Nous ne l'avons pas fait pour deux raisons. La première est qu'il aurait fallu commuter quatre composants, et ce pour l'obtention de cinq fréquences différentes. Cela aurait nécessité d'une part un commutateur très difficile à trouver dans le commerce, et d'autre part la mise en place d'un nombre incroyable de fils de cablage. La deuxième raison qui nous a incité à adopter cette solution est qu'ainsi les cinq oscillateurs restent sous tension et ont le temps de se stabiliser entre l'emploi de fréquences différentes. Vu le nombre restreint de composants de chaque circuit, et leur prix de revient relativement bas, nous ne pensons pas que celà constitue un handicap insurmontable. Cinq fréquences ont été choisies : 10Hz, 100Hz, 1000Hz, 10kHz et 100 kHz. Elles ont été déterminées comme étant les plus employées dans le



Figure 4

domaine des mesures et du test en BF ; mais rien n'est définitif, et d'autres fréquences peuvent être obtenues par échange des condensateurs C1 à C10. Pour exemple, nous avons fait des essais avec des condensateurs de 12pF ; nous avons obtenu une fréquence de plus de 350kHz. Un commutateur connecte la sortie de l'oscillateur choisi à l'entrée de l'étage final constitué par un dernier amplificateur opérationnel monté en suiveur. Chaque AOP a chacune de ses broches connectée aux lignes d'alimentation par une cellule de filtrage constituée d'une résistance de 10 ohms et d'un condensateur de 100 nF.

L'alimentation de l'ensemble est confiée à une alimentation symétrique fournie par deux régulateurs intégrés (LM317 pour la tension positive et LM337 pour la tension négative). Elles seront règlées par les ajustables RV11 et RV12 à +10V et -10V. Le transformateur TR1 est un modèle de 2 X 12V à 2 X 15V. Le redressement est assuré par un pont de diodes (D11), et le filtrage par des condensateurs de 2200µF (tension de service de 35V).



LA REALISATION

Le dessin du circuit imprimé est donné à la **figure 3.** Il est de dimensions respectables, mais supporte tous les composants, y compris l'alimentation et le commutateur de gammes. Nous n'y avons pas inclus le transformateur TR1 bien qu'il soit de taille modeste, afin de limiter les risques d'induction de 50Hz du secteur dans les circuits.

Le cablage sera effectué en s'aidant du dessin d'implantation représenté en **fi-**

gure 4. Bien qu'assez courant, le modèle du commutateur employé pour la réalisation de la maquette pourrait poser des problèmes quant à son approvisionnement. Dans ce cas, le dessin des pistes devra être modifié afin de pouvoir adapter un autre modèle ; ceci ne devrait pas poser de difficultés insurmontables vu le petit nombre de pistes à cet endroit du circuit.
Toutes les résistances ajustables, hormis celles de l'alimentation, seront des modèles multitours pour un règlage plus aisé des oscillateurs.
Il est inutile de prévoir des refroidisseurs pour les deux régulateurs vu le courant faible qu'ils auront à débiter.

Les réglages

Avant de placer les circuits intégrés sur leur support, il faudra règler l'alimentation à une valeur de +10V et -10V. Cela effectué, on pourra vérifier le bon fonctionnement de chaque oscillateur. Il serait étonnant que ces derniers fonctionnent immédiatement sans qu'il soit nécessaire d'agir sur les ajustables de gain RV2, RV4, RV6, RV8 et RV10. Une fois les oscillations démarrées, règler le niveau de sortie à la valeur souhaitée.
Pour ce qui est du règlage de la fréquence, il faudra agir sur les ajustables RV1, RV3, RV5, RV7 et RV9. Si les fréquences annoncées plus haut ne pouvaient, pour certaines d'entre elles, être obtenues, celà laisserait à supposer que les condensateurs de l'oscillateur correspondant ne seraient pas de la valeur requise (tolérance trop grande). Il faudra dans ce cas soit l'augmenter par la mise en parallèle de petites capacités, soit l'échanger contre d'autres valeurs moins élevées.
Quoiqu'il en soit, les règlages de fréquence devront probablement être retouchés après une certaine durée de fonctionnement.

Patrice OGUIC

NOMENCLATURE.

Résistances :
R1, R7, R13, R19, R25 : 10 kΩ
R2, R3, R4, R5, R8, R9, R10, R11, R14, R15, R16, R17, R21, R22, R23, R24, R26, R27, R28, R29 : 100 kΩ
R6, R12, R18, R24, R30 : 1 MΩ
RF : résistances de filtrage 10 Ω X 22 pièces
R31, R32 : 221 Ω 1%

Résistances ajustables :
RV1, RV3, RV5, RV7, RV9 : 4,7 kΩ multitours
RV2, RV4, RV6, RV8, RV10 : 10 kΩ multitours
RV11, RV12 : 4,7 kΩ

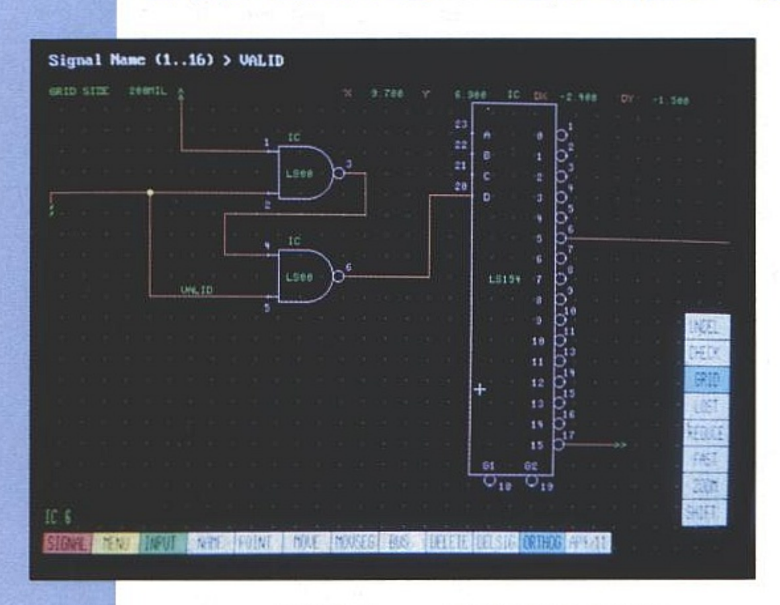
Condensateurs :
C1, C2 : 470 nF
C3, C4 : 47 nF
C5, C6 : 4,7 nF
C7, C8 : 470 pF
C9, C10 : 47 pF
C11, C12, C16, C18 : 100 nF
C13, C14 : 2200 µF 35V
C15, C17 : 2,2 µF 25V
C19, C20 : 10 µF 25V
CF : condensateur de filtrage AOP de 100 nF X 22 pièces

Semi-conducteurs :
D1, D2, D3, D4, D5, D6, D7, D8, D9, D10 : 1N4148
D11 : pont de redressement 500mA ou 1A 50V

Circuits intégrés :
CI1, CI2, CI3, CI4, CI5, CI6, CI7, CI8, CI9, CI10, CI11 : TL071
CI12 : LM317
CI13 : LM337

Divers :
TR1 : transformateur 2 X 15V à 2 X 18V, 4VA
SW1 : commutateur 5 positions à poussoir (BECUWE)
F1 : fusible rapide 200mA

L'ENSEMBLE DE CAO KADS-CAD S3600



La Conception Assistée par Ordinateur est le dernier cri de la conception électronique. De nombreux logiciels destinés à l'élaboration du schéma électronique et à son routage fleurissent ici et là. Parmi toutes les offres disponibles et provenant du monde VAX, la société KADS-CAD propose tout un ensemble de produits dédiés à l'édition d'un schéma électronique et de son routage sur circuit imprimé. C'est une présentation de ce produit que nous vous convions de suivre.

La version qui nous a été confiée a comme référence S3600 en version 1.3 et fonctionne pour la gamme des ordinateurs compatibles IBM PC. Elle se présente sous la forme de trois disquettes 5 1/4 et d'un volumineux manuel, bien aéré tout de même à la différence de la majorité des documentations techniques où l'emploi de la loupe semble de plus en plus obligatoire.
Avant d'entrer dans le vif du sujet, précisons la configuration requise pour ce logiciel. Ne sacrifiant pas à la mode actuelle du tout windows qui - entre parenthèses - n'est pas forcément utile dans le cas d'une CAO, ce logiciel se destine à un environnement DOS (3.2 minimum). De plus, son fonctionnement est fortement dépendant du système d'exploitation et fait partie intégrante de celui-ci. La nécessité d'employer un PC-AT est maintenant toute naturelle, les anciens PC à base de 8088 ou 8086 n'étant plus à même de répondre aux besoins de puissance graphique d'une CAO. Pour cela, une carte graphique EGA (en 640X350) ou VGA (en 640X480) est aussi nécessaire. Notons aussi l'obligation d'un port imprimante, non pas pour vos sorties de contrôle sur papier mais plutôt pour le dongle de protection.
Après l'installation du logiciel, qui se fait de manière fort simple par ailleurs, on retrouve sur le disque dur de l'ordinateur et dans deux répertoires principaux, les exécutables et les données du programme. Parmi celles-ci, on dispose les librairies graphiques nécessaires aussi bien pour l'éditeur de schéma que pour le routeur. La place prise par l'ensemble de ces fichiers se situe pour une installation complète vers 8 Mo. La majorité (6,7 Mo) étant prise par les exécutables.

Lancement de la CAO

Le lancement de la CAO se fait simplement par l'entrée de la commande PCB. Vous vous retrouverez alors dans un mode de commande en ligne semblable à celui du DOS. En effet, comme on l'a suggéré plus haut, un grand nombre de fonctions de trois lettres (40 pour être plus précis) sont accessibles directement au clavier. Vous disposez non seulement de ces fonctions mais aussi de toutes les fonctions du DOS courant et de vos utilitaires préférés.
L'utilisation normale consiste alors à spécifier par l'intermédiaire de la commande JOB nom du développement, le produit en cours de développement. Si celui-ci n'est pas défini, cette commande vous installera le nouveau répertoire ainsi que les données initiales. Une fois cette commande effectuée, diverses commandes comme par exemple l'édition des équipotentielles ou des labels des composants peut se rentrer par les commandes WIR ou EDL. Il faut bien remarquer que toute cette préparation de données destinées au routeur peut se faire manuellement par l'emploi de ces diverses commandes. L'autre moyen consiste soit à utiliser un éditeur de schéma externe comme Orcad et à importer la netlist générée, soit encore à utiliser l'éditeur de schéma fourni et dont nous allons détailler le fonctionnement.

L'éditeur de schéma

A la commande CIR tapée dans l'environnement ligne de commande, KADS-CAD charge l'éditeur de schéma graphique. Sous ce système dont le souris est un des composants absolument principaux, vous avez sur le bas de l'écran une série de menus qui vous permettra de naviguer à travers toutes les possibilités et sous-possibilités du logiciel. Naviguer est bien le terme, car il faut comme dans tout nouveau système vous habituer à cette arborescence pour ne pas vous y noyer. Les diverses



commandes sont très nombreuses et ce qui est un symbole de puissance semblera au premier venu un labyrinthe inextricable. Ceux qui s'en donneront la peine auront alors à leur portée toutes les fonctions habituellement disponibles dans ce type de logiciel. Notons tout de même que le type de sélection dans le menu est bien surprenant au premier abord mais finalement assez pratique. Il suffit en effet de passer dessus avec la souris pour qu'il soit validé, au lieu d'aller appuyer avec le bouton souris dessus.
Les possibilités de cet éditeur de schéma sont à la hauteur de ceux présents sur le marché. La grille est ainsi aisément définissable non seulement en pas mais aussi en position. On peut regretter cependant qu'il n'y ait pas d'auto-panning lors du déplacement de la souris. Les symboles fournis regroupent la majorité des composants classiques tels que résistances, capacités, potentiomètres et aussi certains circuits intégrés de la famille TTL. Ces composants sont regroupés dans un répertoire appelé DEVICE.LIB, tandis que des symboles purement graphiques se placent dans un répertoire SYMBOL.LIB. Deux séries différentes sont fournies avec S3600, un ensemble de composants aux normes américaines et un autre au standard européen. On compte dans chacun de ces deux ensembles une centaine de composants disponibles.

Le routeur

Le routeur est lancé par la commande LAY à partir du mode ligne de commande. L'interface graphique est alors similaire à celle rencontrée dans l'éditeur de schéma (c'est l'avantage d'une solution originaire de la même source que de disposer des mêmes mécanismes, et donc pour l'utilisateur d'un apprentissage moindre). Le passage des données de l'un à l'autre est donc prévu en standard, mais aussi l'importation de netlists en provenance d'autres éditeurs de schémas qui seront utilisés par le routeur. Il suffit par exemple de sélectionner le format CALAY à partir d'ORCAD pour produire les données nécessaires au routeur. Là encore, vous pouvez éditer à partir de la ligne de commande l'ensemble des caractéristiques de votre PCB, comme les références, l'orientation et le placement de vos composants à partir de la ligne de commande par l'instruction EDL ou encore l'édition de vos lignes de routage par la commande LIN. Le contrôle des données est donc entièrement accessible à partir de commandes texte, et vous pouvez les éditer sans faire appel au programme graphique. C'est d'une puissance importante, mais beaucoup moins convivial qu'un programme graphique. Celui-ci dispose du même système de menu que l'éditeur de schémas mais avec une arborescence différente. Il vous donnera accès au chargement et placement de vos composants, au routage et enfin à l'édition directe de toutes les caractéristiques de vos éléments à l'écran. C'est le point intéressant de ce logiciel, qui se situe dans l'édition directe de chaque segment du routage avec le contrôle de sa largeur, position, chemin... Les différentes librairies mises à votre disposition regroupent les composants classiques tels que les empreintes de circuits intégrés au format DIL, les résistances, les condensateurs, les transistors ou encore les connecteurs, mais pas de composants de surface ou autres moins classiques. Là encore, ce sera à l'utilisateur de définir des propres formes. Notons que la réutilisation des nouvelles formes est très simple à utiliser, et bien que cela amène un surcroit de travail au départ, au fil de l'utilisation, la personnalisation des composants permettra un travail plus rapide.
Au point de vue des capacités maximales du logiciel, elles sont tout à fait suffisantes pour la majorité des cas avec une dimension maximale de la carte de 832 mm x 832 mm à la résolution de 0,001 inch, ainsi qu'un nombre maximal de composants de 1024.
Un autorouteur est également fourni, accessible par la commande Route. Il est multi-passe avec routage simultané sur 20 couches et paramétrage du nombre de vias par chevelu, de la grille de recherche ou encore du routage à 45 ou 90 degrés.

■ *Edition des paramètres des composants.*

```
ERP.JOB    User  1                                    KADS-CAD LTD 3600 V1.3
ERP ESSAI                                             13-Jan-1994 15:41

Part Labels Editor V1

 CODE  LABEL     DESIGNATION               CODE  LABEL     DESIGNATION
   0    _IC    *Integrated Circuit          16    RL     *Relay
   1    T      *Transistor                  17    F      *Fuse
   2    D      *Diode                       18    L      *Choke
   3    Z      *Zener Diode                 19    TR     *Transformer
   4    LD     *L.E.D.                      20    XT     *Crystal
   5    ??                                  21    ??
   6    ??                                  22    ??
   7    ??                                  23    ??
   8    C      *Capacitor                   24    SW     *Switch
   9    ??                                  25    W      *Strap
  10    R      *Resistor                    26    J      *Connector
  11    RN     *Network                     27    TP     *Test Point
  12    P      *Potentiometer               28    ??
  13    ??                                  29    ??
  14    ??                                  30    ??
  15    ??                                  31    ??     Invalid Label
 E)xit   Q)uit   >
```



```
KADS-CAD LTD 3600 V1.3      Automatic Router Version RV3.X

Parameter File : LOGIC

Maximum Line Deviation ------ ( Mil ) --       75    75   100   200
Maximum Traverse Over-Shoot - ( Mil ) --        0   600   600   600
Maximum Channel Width ------- ( Mil ) --      100   100   200   400
Minimum Traverse Segment ---- ( Mil ) --        0   100   100   100
Maximum No Traverses/Route -- ( 0-6 ) --        0     2     4     6
Search Grid interval -------- ( Mil ) --       50    50    25    25
Minimum Clearance Gap ------- ( Mil ) --       11    11    11    11

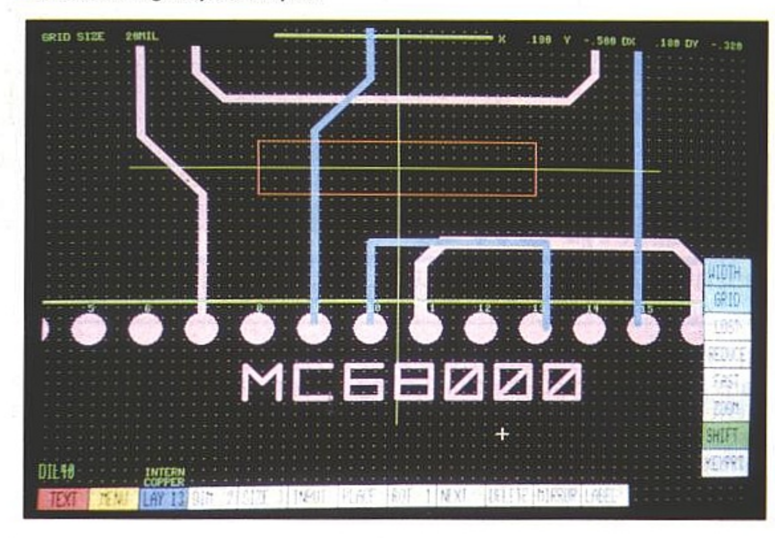
Time per Pass --------------- ( Min ) --        0     0     0     0
No of Routes Done ----------------------        0     0     0     0

No of Horizontal Layers   1        Router Source Layer         46
No of Vertical Layers     1        Traverse Grid Lock         OFF
Horizontal Layer(s) ----- 2        Maximum Diagonal   (Mil)    40
Vertical Layer(s) ------- 1        Route Time Limit   (Min)     5
Current Pass No --------- 1        Alternative Power Code       7
Maximum Pass No --------- 4        Max Reduced Length (Mil)  1000

G)o  A)rtwork  I)nit  R)ead  W)rite  E)xit > _
```

■ *Les paramètres de l'autorouteur*

■ *Edition en gros plan du pcb.*



En outre

Une fois l'ensemble du schéma et du routage terminé, vous pouvez lancer tout un ensemble d'instructions à partir de la ligne de commande. Outre les sorties Plotter sur traceur papier par la commande Plot, vous pouvez vérifier et tester vos équipotentielles, boîtiers, liaisons et composants par les commandes CHEck ou DATacheck. le contrôle des isolements peut lui se faire par l'instruction GAP, ou encore on peut appeler une analyse de la densité de placement par la commande HISto-gram. En cas d'erreur ou de modification éventuelle, on peut relancer un nouveau cycle CIR et LAY. Une fois votre travail vérifié et validé, les sorties photoplotter sont accessibles par les commandes PHOto et EDS qui génèrent successivement les données du phototraçage et les codes associés. La commande DRILL permet quant à elle la sortie des données de perçage pour une perceuse numérique automatique. Une autre commande (OPTimise) permet l'optimisation de vos sorties DRILL et PHOto. Tous ces paramètres de sortie peuvent être édités là encore par une commande SIZes, donnant accès à la table des pastilles, pistes et trous de perçages.
Comme vous le voyez, le nombre de commandes est important, et surtout permettent l'accès à vraiment toutes les données de votre étude. Il s'agit là d'un des points forts du logiciel.

CONCLUSION

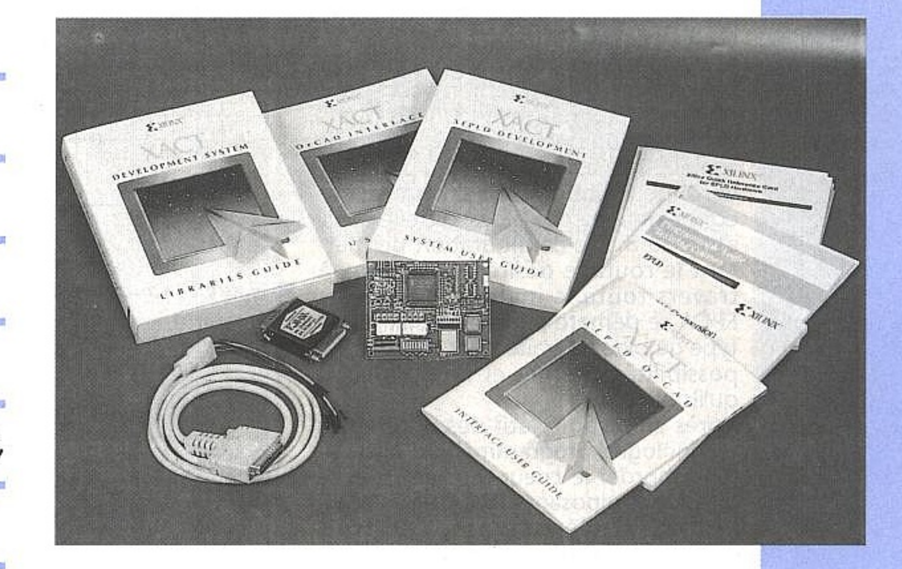
Ce logiciel répond aux besoins de conception électronique de manière certaine. Sa philosophie d'utilisation est cependant assez lourde à gérer. Ceux qui ont l'habitude d'un mode graphique total risquent d'être fortement perturbés par la philosophie essentiellement textuelle de ce programme de CAO. Mais l'avantage se trouve dans la puissance de ces nombreuses commandes. Les potentialités de ce produit sont importantes, mais la prise en main demandera un investissement en temps important pour pouvoir toutes les utiliser avec leur finesse. Une fois ce travail entrepris, vous aurez de quoi produire même vos circuits les plus complexes.
La version présentée est référencée S3600 1.3 avec éditeur de schémas et routeur et est disponible à un prix de 14990 F HT. Une version réduite est cependant prévue pour la fin du premier trimestre. Celle-ci disposera des mêmes fonctionnalités mais avec un nombre de couches et de données plus réduit. Notons aussi qu'à lorigine développé pour station VAX, ce logiciel reste également disponible pour ce système.

KADS-CAD est distribué par :
ADC Informatique
12, route de Chevennes
74960 Cran-Gévrier
Tél. : 50.57.43.63

P. de CARVALHO

LE KIT DE DÉVELOPPEMENT POUR FGPA XILINX

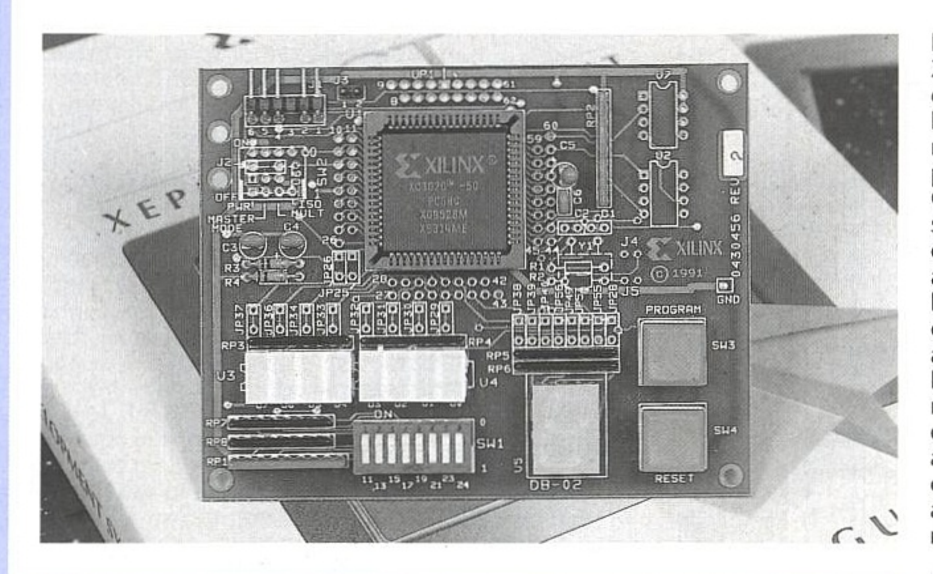
Les composants programmables sont un des fers de lance de l'électronique moderne. Ils disposent d'infiniment plus d'avantages aussi bien dans la conception d'un produit que lors de sa production ou sa maintenance. En effet, que ce soit des microprocesseurs, microcontroleurs, mémoires RAM, DRAM, ROM, EPROM ou encore PAL,GAL, EPLD ou FPGA, ils sont de plus en plus présents dans toute conception. Bien qu'ils soient prévus pour des traitements numériques, il est de plus en plus courant de les trouver dans des applications analogiques, car il y a toujours une partie numérique pouvant potentiellement faire appel à ce type de composants. L'étude que nous vous proposons ici est dédié à l'un des ténors du marché, XILINX. Cette présentation nous conduira dans de prochains numéros à des applications autour de ces composants.



L'usage de certains composants programmables s'est pratiquement banalisé, notamment les microcontroleurs et mémoires dont vous pouvez trouver de nombreux exemples dans tous les journaux liés à l'électronique, à commencer par celui que vous tenez dans la main. L'utilisation de ces composants se résume plus à une philosophie informatique dans laquelle vous devez décrire le processus à effectuer qu'à une approche de type électronique dans laquelle vous connectez (le mieux possible !) des composants existants. L'évolution aidant, ce type même de conception est remise en cause; en effet, vous devez connaitre l'existence des PAL purement combinatoires dans lesquelles les connexions sur une matrice de portes OU ou ET sont programmables. Il vous suffit de penser votre circuit avec des portes logiques standards pour que, après une compilation du fichier décrivant vos équations, vous puissiez programmer le PAL répondant à votre brochage. Les GAL et PAL à registres comportent non seulement la matrice permettant la logique combinatoire mais aussi des blocs configurables contenant généralement un registre. Ce type de composants vous permet déjà de former de petites machines à états comme des compteurs ou des accumulateurs. Là encore, il vous suffit de décrire votre système par un ensemble de portes logiques et de registres, pour que, muni de la bonne interface, vous puissiez programmer votre composant suivant vos désirs.

Ces composants que nous vous avons présentés sont connus et présentent tous généralement une structure du type de la **figure 1**. Malheureusement, une étude électronique s'accommode assez mal d'une telle structure qui ne comporte qu'un seul étage et généralement formée de peu de blocs. Lors d'une conception standard, votre structure ressemble plus à celle de la **figure 2**, dans laquelle les connexions entre composants se répartissent sur toute la surface du circuit. Pour satisfaire à ce type d'application, les FPGA re-





■ *Figure 9 : exemple d'auto-programmation par configuration parallèle.*

Figure 10 : exemple d'une décomposition de schéma en éléments simples du FGPA

me XACT. En effet, celui-ci aura introduit ses composants dans votre «library», et vous devrez concevoir votre étude avec ceux-ci. Mais rassurez vous, la majeure partie d'entre eux ne sont que des représentations de composants classiques, comme des circuits TLL (LS138,LS74) ou encore des composants génériques (Multiplexeur 2,3 entrées, portes AND,OR à n entrées,...).
Toute cette partie se déroule de la plus habituelle des façons. Par contre, une fois ce travail achevé, il vous faudra passer à la production des fichiers nécessaires aux XILINX.

Le pont ORCAD/XACT.

Une fois établi votre schéma d'étude, il vous faut donc produire vos données de programmation. Pour cela, le pont livré se trouve dans un programme XMAKE. Celui-ci en fonction du fichier d'entrée sélectionne le bon pont et délivre tous les fichiers nécessaires. Il rentre dans tous les sous-schémas pour produire aussi les fichiers appropriés. Ce programme déclenche toute une série d'écutables automatiquement, dans laquelle l'utilisateur n'a plus qu'à attendre le résultat final. On a dans l'ordre : l'éxecution du convertisseur fichiers schémas, ensuite le placement automatique des différents blocs avec leur routage (programme APR que nous verrons plus loin), et enfin la production des fichiers de programmation au format bitstream. Ce programme constitue donc un séquenceur des autres commandes disponibles, que nous allons voir plus en détail.

Le convertisseur de schéma

C'est à ce convertisseur que revient le rôle de lire votre fichier schéma et de le traduire en un ensemble de blocs de type CLB et IOB avec leurs connexions associées. Pour cela, ce convertisseur utilise un ensemble pour traduire les fonctions de haut niveau provenant de l'éditeur de schéma comme les compteurs binaires, les comparateurs et autres en un ensemble d'éléments compréhensibles par la structure du FPGA (**figure 10**). Le fichier produit a comme extension LCA et contient l'ensemble des données originales de votre schéma mais représentées par un ensemble de IOBs, CLBs et nets. Il faut bien remarquer que cet ensemble de blocs est un ensemble flottant, sans attache physique. Ce rôle est dévolu au programme que nous allons voir.

Le programme XNFDRC.

Cette étape est optionnelle mais fortement recommandée !. Il s'agit d'un programme qui lance une série de tests et qui s'assure de la validité ou de la «faisabilité» de votre étude. A partir d'un fichier XNF produit par le convertisseur de schéma, il teste les incomptabilités de ressources comme un nombre trop important d'horloges ou de blocs (IOB ou CLB) par rapport à ce qui est disponible. En outre, il identifie des structures à même de causer des problèmes comme des éléments avec des FAN-OUTs importants, des "lignes longues" avec peu de composants rattachés, ..
Il propose aussi certaines solutions à des problèmes déjà résolus ou programmés. Cette étape est comme vous le voyez assez importante,pas forcement obligatoire, mais certainement très utile.

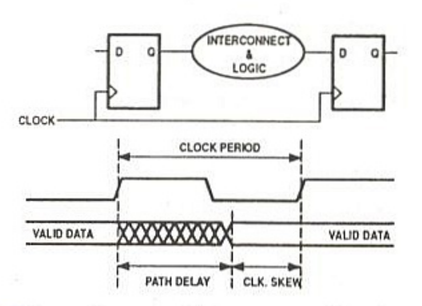
L'APR.

Sous cet acronyme se cache l'un des composants principaux du système de développement. En effet, c'est à l'APR (Automatic Place and Route) que revient le rôle de décider de la structure de votre matrice de CLB avec ses connexions. Car jusqu'à ce point là, les éléments n'étaient considérés que

comme une structure FPGA non rattachée à un composant physique. A partir du fichier généré par l'APR, la structure a été inculquée à un composant physique comme par exemple un XC2064-70PC44 désignant un FPGA de type 2064 dans un boitier PLCC à 44 broches et avec une fréquence de fonctionnement de 70MHz. Pour ce faire, le programme, considère l'ensemble des blocs fourni dans le fichier LCA de départ et essaye de placer chaque bloc dans un bloc du composant. Diverses options sont possibles à ce point du traitement. Car certaines contraintes peuvent être imposées aussi bien au placement qu'au routage des connexions. On peut obliger ou prohiber telle entrée/sortie sur un IOB, ou encore bloquer ou interdire un groupe de CLB. La priorité des connexions et différents nets est également configurable, pour des questions de temps de propagation ou encore de ressources internes. Le routage des connexions se fait alors entre ces différents blocs, pour produire en final un fichier LCA satisfaisant. Un programme référencé APRLOOP peut récursivement faire appel à l'APR pour essayer différents types de configuration, lorsque le résultat produit par l'APR est insuffisant. Ceci se produit généralement lorsque les ressources internes du FPGA commencent à être bien remplies (80%) et que l'aide apportée par l'utilisateur à l'APR, par l'intermédiaire du fichier de contraintes, est difficile à gérée comme lorsque la compilation découle d'un éditeur de schémas. Au cas où l'APR arrive à caser l'ensemble de vos blocs et connexions dans le composant sélectionné, il vous produit un autre fichier LCA mais qui là contient bien la bonne structure rattachée à votre composant. A partir de celui-ci, il est possible de produire le fichier de programmation nécessaire au composant et ceci par l'intermédiaire d'un autre programme de fin de chaine.

Le programme MAKEBIT et MAKEPROM

La fin de la chaine de traitement est proche, et est représentée par le programme MAKEBIT. Comme son nom l'indique, il a pour charge de produire le fichier bitStream qu'accueillera la mémoire de configuration du FPGA sélectionné. Pour cela, il va traduire le fichier LCA produit par l'APR, en un ensemble de points de commutation chargés de selectionner de manière interne telle ou telle configuration. Le fichier avec pour extension bit fourni en sortie, est le fichier qu'il faudra programmer dans la ROM série associée au FPGA. Ce fichier peut également servir à un autre programme nommé MAKEPROM qui, lui, produira les données nécessaires à la programmation de PROM standard au format parallèle. La possibilité est aussi offerte à MAKEBIT de produire les différents temps de propagation des routes ainsi générées pour un usage de vérification que nous verrons plus loin.



■ *Figure 11 : exemple de rapport donné par XDelay.*

L'interface XDM.

Tous les programmes que l'on a pu énumérer plus haut, et aussi l'XDE que l'on va voir, peuvent être lancés non seulement à partir de la ligne de commande, mais aussi à partir d'un intégrateur graphique. Celui-ci appelé XDM (XACT Design Manager) vous permet d'avoir sous la main tous les programmes, avec commande à la souris et rappel de toutes les options de chaque programme. Une aide en ligne est également disponible, qui vous permettra de rapidement situer au départ, la fonctionnalité de chaque élément. Notons qu'il est également possible de lancer votre éditeur de schéma à partir de celui-ci. Il vient s'installer dans le menu DesignEntry.

Le module XDE.

Ce programme qui n'est pas livré avec le produit que nous avons reçu, et donc que nous n'avons pas pu tester, est une autre alternative au développement à partir de l'éditeur de schéma, ou plutôt un complément à celui-ci. Le XDE (XACT Design Editor) permet de visualiser, modifier directement la matrice de CLBs, avec toutes ses interconnexions. Comme vous pouvez l'imaginer, ce travail pour puissant qu'il paraisse est notablement difficile. Il faut en effet router, configurer chaque CLB, IOB, connexion à travers la structure en n'oubliant pas les contraintes de temps, de placement et de connexions. Par contre, vous accédez réellement à toutes les possibilités du FPGA, par l'édition directe de ses composantes. Cet outil est nettement intéressant dans le cas d'un routage critique en temps de propagation, où tout routeur même pour FPGA, est loin de l'intellect humain. On pourra donc concevoir l'étude avec un éditeur de schéma classique, produire les fichiers XILINX, et ensuite rerouter les signaux critiques.

La vérification

Toute la séquence que nous vous avons décrite plus haut est la séquence parfaite, c'est à dire que tout se passe bien, sans la moindre erreur aussi bien de conception, que de placement ou routage. Malheureusement, l'erreur zéro n'existant pas, une partie du système de développement est prévue pour la vérification de votre étude. La première vérification s'éffectue lors du cycle de production du fichier bitstream. Si les ressources internes du FPGA sont insuffisantes, alors un message d'erreur vous alerte du problème constaté.

Une autre source d'erreur vient du caractère spécial des différents types de connexions inclus dans un FPGA. En effet chaque connexion disposant d'un temps de propagation différent, il peut s'avérer qu'il puisse s'introduire un certain nombre de glitches, ou encore qu'il faille valider des registres après le passage du signal utile.

Pour une vérification de ces différents temps de propagation, le programme XDELAY permet de créer une liste détaillée de tous les temps de parcours, et produit une analyse du circuit avec l'estimation des performances pour chaque horloge de la structure (**figure 11**).

Un programme LCA2XNF permet à partir du fichier LCA programmé dans le composant, de produire les données nécessaires pour un simulateur logique (comme ceux de ViewLogic, Orcad ....). Il vous permet de resimuler votre étude avec les temps de propagation des composants FPGA. Un autre programme dénommé BAX (pour Back Annotation Program) permet la rétro-annotation dans le schéma original, afin de visualiser à partir du simulateur logique les temps de propagations, mais directement dans le schéma de départ.

Le hardware

Le kit de développement est fourni avec une petite plaque d'essais comportant entre autres un support pour FPGA de la série XC3020 ou XC2064, associé à un ensemble d'afficheurs, LED, interrupteurs en boitier DIL, et boutons poussoir. Ce mini-kit est destiné à la pratique de ces FPGA qui n'est pas dans la philosophie immédiate de tout électronicien. La programmation se fait directement à partir d'un câble relié sur le port série d'un PC. Tout cycle de programmation (E)PROM est donc évité et permet une visualisation directe de votre test.

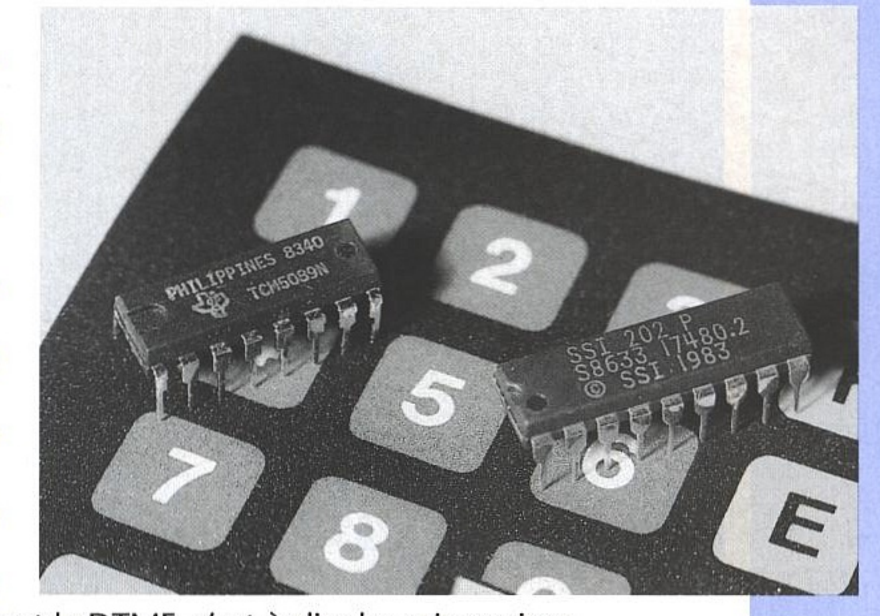
Conclusion

Comme vous avez pu le remarquer, cet ensemble de développement se veut résolument professionnel. L'utilisation des FPGA nécessite des outils appropriés, mais autorise d'énormes possibilités. Cela ne veut pas dire inaccessible, et nous aurons certainement l'occasion de vous fournir dans de prochains numéros des applications utilisant de tels composants. Ne vous effrayez pas, il ne vous faudra pas obligatoirement posséder le système de développement, un simple programmateur d'EPROM tout bête sera suffisant. Nous vous fournirons en effet le fichier binaire nécessaire à la production de la PROM de configuration . Comme vous le voyez, nous ne ménageons pas nos efforts pour vous informer sur les technologies les plus récentes actuellement sur le marché.

P. de CARVALHO

UN CIRCUIT DE DÉCODAGE POUR COMMANDES DTMF

Les tonalités audibles codées sont extrêmement utilisées pour la télécommande et l'identification à distance via des supports de communication prévus pour la parole (réseau téléphonique commuté, voies radio, ou même cassettes magnétiques).



Le codage le plus populaire est évidemment le DTMF, c'est-à-dire les seize paires de fréquences des claviers téléphoniques modernes, mais il ne faut pas oublier les systèmes à fréquences simples utilisés en appel sélectif (ZVEI, CCIR, etc.)

La reconnaissance des tonalités proprement dites est généralement confiée à des composants spécialisés, mais celle des «codes confidentiels» propres à chaque application doit être traitée à part.

A mi-chemin entre la «serrure» en logique câblée et le microcontrôleur spécialement programmé, la réalisation d'un composant spécifique à partir d'un PAL ou d'un EPLD simple s'avère particulièrement intéressante.

Reconnaître des suites de quartets

La plupart des circuits intégrés récepteurs de tonalités délivrent un code à quatre bits (autrement dit un quartet) pour chacune des seize combinaisons (ou moins) qu'ils sont amenés à reconnaître. Toute identification d'une combinaison valide, quelle qu'elle soit, est par ailleurs signalée par le changement d'état d'une broche séparée (DV).
A titre d'exemple, la **figure 1** reproduit une affectation de ces quatre bits couramment utilisée dans les décodeurs DTMF (par exemple le SSI 202 ou SSI 75T202 de SILICON SYSTEMS, bien connu de nos lecteurs).
On remarquera qu'il diffère quelque peu du BCD ou de l'hexadécimal courant, notamment au niveau du codage de la touche «zéro».
Le circuit intégré que nous nous proposons de créer, dont la **figure 2** schématise le brochage, devra être capable de reconnaître deux codes distincts, composés chacun de trois quartets consécutifs.
Ramené au niveau d'un clavier téléphonique, cela correspondra tout simplement à la composition de deux codes à trois chiffres.
L'un pourra être affecté à une fonction «marche» et l'autre à une fonction «arrêt», mais tout autre usage est bien entendu possible (par exemple la transmission de deux signaux différents en radiomessagerie simple).
Outre les quatre entrées et leur horloge «DV» pour les données en provenance du CI récepteur, notre composant PG3030 possèdera naturellement deux sorties A et B (actives au niveau haut), deux entrées séparées pour la remise à zéro des sorties A et B, et deux entrées de validation de ces mêmes sorties (elles aussi actives au niveau haut).
Moyennant des interconnexions



PG3060
PAL 16R6
(CMOS)

+5V

1 DV, 2 D1, 3 D2, 4 D4, 5 D8, 6 OEA, 7 OEB, 8 RAZA, 9 RAZB, 10 GND
20 VCC, 19 SA, 18, 17, 16, 15, 14, 13, 12 SB, 11

SA (Sortie active au niveau haut)
TEST A — Sortie A : 7 6 5 — Sortie B : 4 5 6
TEST B — TEST
SB (Sortie active au niveau haut)
(OE sorties de test)

Figure 2

```
title (SERRURE DTMF)
include p16r6;
define b0= (!pin4 & !pin5);
define b4= (!pin3 & !pin2);
define b5= (!pin3 & pin2);
define b6= (pin3 & !pin2);
define b7= (pin3 & pin2);
pin19.oe= pin6;
pin12.oe= pin7;
!pin12= pin13.q | pin14.q | pin15.q;
!pin19= pin16.q | pin17.q | pin18.q;
state_diagram !pin13, !pin14, !pin15
{
if (b0 | pin5 | !pin9) s0;
state s0=000
if (b7) s1; else s0;
state s1=001
if (b6) s2; else s1;
state s2=011
if (b5) s3; else s2;
state s3=111
s1;
}
state_diagram !pin16, !pin17, !pin18
{
if (b0 | pin5 | !pin8) s0;
state s0=000
if (b4) s1; else s0;
state s1=001
if (b5) s2; else s1;
state s2=011
if (b6) s3; else s2;
state s3=111
s1;
}
```

Figure 3

```
pin19.oe=
          pin6

pin12.oe=
          pin7

!pin12=
          pin13.q
        | pin14.q
        | pin15.q

!pin19=
          pin16.q
        | pin17.q
        | pin18.q

pin13.d=
          pin13.q & pin2 & pin14.q & pin9 & !pin5 & pin4
        | pin13.q & pin3 & !pin15.q & pin9 & !pin5 & pin4
        | pin13.q & !pin2 & !pin15.q & pin9 & !pin5 & pin4
        | pin13.q & !pin2 & pin14.q & pin15.q
        | !pin4
        | !pin9
        | pin5

pin14.d=
          pin14.q & !pin3 & pin13.q & pin9 & !pin5 & pin4
        | pin14.q & pin2 & pin13.q & pin9 & !pin5 & pin4
        | pin14.q & !pin2 & pin13.q & pin15.q
        | !pin4
        | !pin9
        | pin5

pin15.d=
          pin15.q & pin13.q & pin14.q & !pin2
        | pin15.q & pin13.q & pin14.q & !pin3
        | !pin4
        | !pin9
        | pin5

pin16.d=
          pin16.q & !pin2 & pin17.q & pin8 & !pin5 & pin4
        | pin16.q & !pin3 & !pin18.q & pin8 & !pin5 & pin4
        | pin16.q & pin2 & !pin18.q & pin8 & !pin5 & pin4
        | pin16.q & pin2 & pin17.q & pin18.q
        | !pin4
        | !pin8
        | pin5

pin17.d=
          pin17.q & pin3 & pin16.q & !pin18.q & pin8 & !pin5
        | pin17.q & !pin2 & pin16.q & pin8 & !pin5 & pin4
        | pin17.q & pin2 & pin16.q & pin18.q
        | !pin4
        | !pin8
        | pin5

pin18.d=
          pin18.q & pin16.q & pin17.q & pin2
        | pin18.q & pin16.q & pin17.q & pin3
        | !pin4
        | !pin8
        | pin5
```

Figure 4

Digit	Hexadecimal D8	D4	D2	D1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	1	0	1
0	1	1	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

Figure 1

simples, cela permettrait entre autres possibilités de faire interagir les deux sorties l'une sur l'autre selon des modalités variées, ou tout simplement de procéder à une RAZ «propre» lors de la mise sous tension (à l'aide d'un simple réseau RC).

Six broches «de test» commandées par une entrée de validation indépendante (/OE) pourront servir à visualiser le cycle décrit par les six bascules du PAL 16R6 servant de base à notre développement.

Nous avons en effet choisi d'affecter une bascule individuelle à chacune des trois étapes de la composition de chacun des deux codes, et une fonction combinatoire à la détection d'un triple niveau haut sur chaque groupe de bascules.

La capacité du 16R6 est ainsi bien rentabilisée, ce qui n'est pas si fréquent en matière de PAL!

Si on fait coïncider les poids binaires des entrées du PG3030 et du récepteur DTMF, le code de la sortie A est «765» et celui de la sortie B «456».

Il n'est toutefois nullement interdit de «croiser» certaines lignes de données, pour arriver par exemple à «206» et «B60», ou «153» et «735», ou encore «236» et «763».

Evidemment, toute réception d'un chiffre ne faisant pas partie de l'un ou l'autre des deux codes remet l'ensemble des bascules à zéro, assurant une sécurité que l'on peut considérer comme satisfaisante pour les applications les plus courantes.

Développement sous Prologic

Partant de ce cahier des charges, il est particulièrement commode d'utiliser un «compilateur logique» pour développer le schéma que l'on programmera dans le PAL 16R6 pour en faire un PG3030.

Rappelons que ce logiciel, dont une version limitée à ses propres références a été diffusée gratuitement par TEXAS INSTRUMENTS (nous l'offrons maintenant en téléchargement), accepte une description du système logique à créer sous la forme de tables de vérité, d'équations booléennes, ou de diagrammes d'état.

Dans notre cas, c'est une combinaison de ces deux dernières techniques qui nous a servi à écrire le «code-source» de la **figure 3**.

Il pourra éventuellement être modifié par ceux de nos lecteurs dont les besoins différeraient peu ou prou de ce que peut faire le PG3030: c'est l'un des gros avantages des «réseaux logiques programmables»...

Sauvegardé sur disque sous le nom «SERDTMF.PLD», il pourra être soumis à PROLOGIC en tapant simplement «LC SERDTMF».

Le résultat de ce traitement est le jeu d'équations «minimisées» de la **figure**



```
proLogic Compiler
Texas Instruments V1.97
Copyright (C) 1989 INLAB, Inc.
Fuse Plot

        SERRURE DTMF

p16r6 revision 89.2.11

                11 1111 1111 2222 2222 2233
   0123 4567 8901 2345 6789 0123 4567 8901

 0 ---- ---- ---- ---- X--- ---- ---- ----  OE
 1 ---- ---- ---- ---X ---- ---- ---- ----  +
 2 ---- ---- ---X ---- ---- ---- ---- ----  +
 3 ---- ---X ---- ---- ---- ---- ---- ----  +   !pin19
 4 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
 5 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
 6 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
 7 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

 8 X--- ---X ---X ---X ---- ---- ---- ----  +
 9 ---- X--X ---X ---X ---- ---- ---- ----  +
10 ---- ---- -X-- ---- ---- ---- ---- ----  +
11 ---- ---- ---- ---- ---- ---- -X-- ----  +   pin18.d
12 ---- ---- ---- X--- ---- ---- ---- ----  +
13 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
14 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

16 ---- X-X- ---X -X-X ---- ---- X--- ----  +
17 -X-- ---- X--X -X-X ---- ---- X--- ----  +
18 X--- ---X ---X ---X ---- ---- ---- ----  +
19 ---- ---- -X-- ---- ---- ---- ---- ----  +   pin17.d
20 ---- ---- ---- ---- ---- ---- -X-- ----  +
21 ---- ---- ---- X--- ---- ---- ---- ----  +
22 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

24 -X-- ---- X--X -X-X ---- ---- X--- ----  +
25 ---- -XX- X--- -X-X ---- ---- X--- ----  +
26 X--- --X- X--- -X-X ---- ---- X--- ----  +
27 X--- ---X ---X ---X ---- ---- ---- ----  +   pin16.d
28 ---- ---- -X-- ---- ---- ---- ---- ----  +
29 ---- ---- ---- ---- ---- ---- -X-- ----  +
30 ---- ---- ---- X--- ---- ---- ---- ----  +
31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

32 -X-- ---- ---- ---- ---X ---X ---X ----  +
33 ---- -X-- ---- ---- ---X ---X ---X ----  +
34 ---- ---- -X-- ---- ---- ---- ---- ----  +
35 ---- ---- ---- ---- ---- ---- ---- -X--  +   pin15.d
36 ---- ---- ---- X--- ---- ---- ---- ----  +
37 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

40 ---- -X-- X--- -X-- ---- ---X ---X X---  +
41 X--- ---- X--- -X-- ---- ---X ---X X---  +
42 -X-- ---- ---- ---- ---X ---X ---X ----  +
43 ---- ---- -X-- ---- ---- ---- ---- ----  +   pin14.d
44 ---- ---- ---- ---- ---- ---- ---- -X--  +
45 ---- ---- ---- X--- ---- ---- ---- ----  +
46 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
47 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

48 X--- ---- X--- -X-- ---- ---X ---X X---  +
49 ---- X--- X--- -X-- --X- ---- ---X X---  +
50 -X-- ---- X--- -X-- --X- ---- ---X X---  +
51 -X-- ---- ---- ---- ---X ---X ---X ----  +   pin13.d
52 ---- ---- -X-- ---- ---- ---- ---- ----  +
53 ---- ---- ---- ---- ---- ---- ---- -X--  +
54 ---- ---- ---- X--- ---- ---- ---- ----  +
55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

56 ---- ---- ---- ---- ---- X--- ---- ----  OE
57 ---- ---- ---- ---- ---- ---- ---X ----  +
58 ---- ---- ---- ---- ---- ---X ---- ----  +
59 ---- ---- ---- ---- ---X ---- ---- ----  +   !pin12
60 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
61 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
62 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +
63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX  +

   : :  : :  : :  : :  : :  : :  : :  : :
pin2 :  3 :  4 :  5 :  6 :  7 :  8 :  9 :
     :    :    :    :    :    :    :    :
   pin1   18   17   16   15   14   13   11

Legend:

  X  : Cell intact     (JEDEC 0)
  -  : Cell programmed (JEDEC 1)

  X- : True input term, Complement register term
  -X : Complement input term, True register term
  XX : Any XX pair in a product term yields product term LOW.
  -- : No input term (don't care).  A product term comprised
       entirely of -- yields product term HIGH.
```

Figure 5

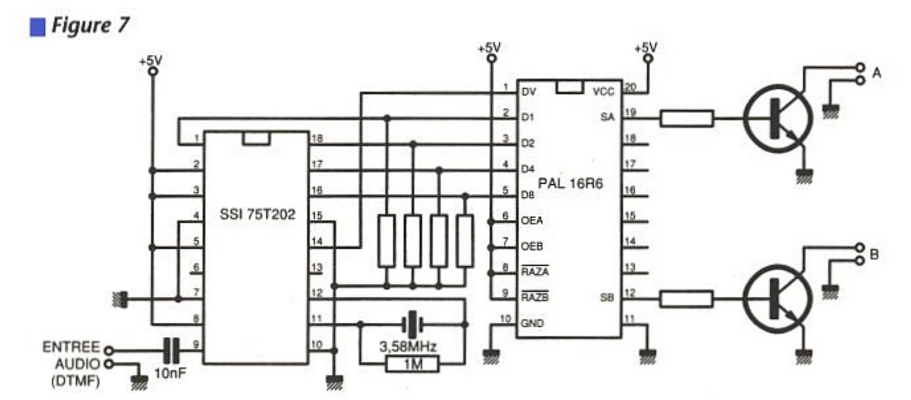


Figure 7

```
proLogic Compiler
Texas Instruments V1.97
Copyright (C) 1989 INLAB, Inc.

        SERRURE DTMF

p16r6 revision 89.2.11

*N_csidp16r6
*QP20
*QF2048
*F0
*L0000 11111111111111111101111111111111
*L0032 11111111111111111011111111111111
*L0064 11111111111011111111111111111111
*L0096 11111110111111111111111111111111
*L0256 01111110111011101111111111111111
*L0288 11110110111011101111111111111111
*L0320 11111111101111111111111111111111
*L0352 11111111111111111111111111011111
*L0384 11111111111110111111111111111111
*L0512 11110101111010101111111110111111
*L0544 10111111101101010111111110111111
*L0576 01111110111011101111111111111111
*L0608 11111111101111111111111111111111
*L0640 11111111111111111111111111011111
*L0672 11111111111110111111111111111111
*L0768 10111111101101010111111110111111
*L0800 11111001011110101111111110111111
*L0832 01111101011110101111111110111111
*L0864 01111110111011101111111111111111
*L0896 11111111101111111111111111111111
*L0928 11111111111111111111111111011111
*L0960 11111111111110111111111111111111
*L1024 10111111111111111110111011101111
*L1056 11111011111111111110111011101111
*L1088 11111111101111111111111111111111
*L1120 11111111111111111111111111111011
*L1152 11111111111110111111111111111111
*L1280 11111011011110111111111011100111
*L1312 01111111011110111111111011100111
*L1344 10111111111111111110111011101111
*L1376 11111111101111111111111111111111
*L1408 11111111111111111111111111111011
*L1440 11111111111110111111111111111111
*L1536 01111111011110111111111011100111
*L1568 11110111011110111101111111100111
*L1600 10111111011110111101111111100111
*L1632 10111111111111111110111011101111
*L1664 11111111101111111111111111111111
*L1696 11111111111111111111111111111011
*L1728 11111111111110111111111111111111
*L1792 11111111111111111111101111111111
*L1824 11111111111111111111111111101111
*L1856 11111111111111111111111110111111
*L1888 11111111111111111110111111111111
*C9F2D
*V7A3D
```

Figure 6

4, et le «plan des fusibles» de la **figure 5**.
Il reste donc à câbler ce schéma au moyen d'un programmateur approprié.

Le transfert sur silicium

Le fichier JEDEC de la **figure 6** (SERDTMF.JED) permet de programmer n'importe quelle version de PAL 16R6: depuis le TIBPAL 16R6 (bipolaire) jusqu'au TICPAL 16R6 (CMOS) de TEXAS INSTRUMENTS, en passant par le PALC 16R6 de CYPRESS (distribué par NEWTEK) si on souhaite utiliser le programmateur ultra-simple dont les plans se trouvent dans notre ouvrage COMPOSANTS ELECTRONIQUES PROGRAMMABLES (ETSF éditeur).
Mais on peut aussi faire appel à des EPLD universels à 20 broches, pourvu qu'ils soient capables d'émuler le 16R6.
Tel est le cas, entre autres, du PEEL 18CV8 d'ICT, qui peut être programmé avec le fichier JEDEC SERDTMF.JEX et de l'EP320 d'ALTERA qui nécessite pour sa part le fichier JEDEC SERDTMF.JEA, tous deux disponibles sur notre serveur.
Il ne reste maintenant plus qu'à mettre à contribution ce composant exclusif, par exemple dans un schéma inspiré de celui de la **figure 7**.
Mais attention, les PAL sont des composants très rapides: il faudra soigner tout particulièrement les découplages!

Patrick GUEULLE



à partir de 16 500 F*

A ce prix là, les nouveaux TDS 300 ont bien quelques concurrents, mais pas d'adversaire.

Pour le même prix existe-t-il d'autres oscilloscopes disposant d'une telle vitesse d'échantillonnage ? A part les nouveaux TDS 300, trouve-t-on sur le marché des oscilloscopes numériques offrant une bande passante aussi large en répétitif et en monocoup ? Non, aucun. Pourquoi ? Parce que les TDS 300 de Tektronix sont les premiers oscilloscopes numériques temps-réel de leur catégorie. Ils vous donnent une confiance totale dans la représentation des signaux, sans qu'il soit nécessaire de repasser en mode analogique comme avec les solutions combinées. Avec les TDS 300, vous disposez de 21 mesures automatiques, d'une option recopie d'écran et de toutes les possibilités de programmation du numérique. De plus, leur écran VGA apporte une finesse de trace supérieure à l'analogique. Mais les TDS 300 c'est aussi la fiabilité (20 000 heures de MTBF et une garantie de 3 ans). Aussi familiers que votre oscilloscope analogique, vous les utiliserez sans manuel. Alors, pour ce prix là n'hésitez pas, offrez-vous la technologie Tektronix en appelant : N° Vert 05.00.22.00

	Nouveau TDS 310	TDS 320	Nouveau TDS 350
Bande passante	50 MHz	100 MHz	200 MHz
Vitesse échantillonnage.	200 Mé/s	500 Mé/s	1 000 Mé/s
Prix*	16 500 F	22 900 F	31 500 F



TDS 300 : LES 1er OSCILLOSCOPES NUMÉRIQUES TEMPS-RÉEL.



